



Logger

*The Data Recorder and Player
for the Virtual World*

M Ä K T E C H N O L O G I E S



Logger

*The Data Recorder / Player
for the Virtual World*



*MäK Technologies
185 Alewife Brook Parkway
Cambridge, MA 02138 USA*

Copyright 1997, MäK Technologies
All rights Reserved. Printed in the United States.
First Printing: March, 1997
Under copyright laws, no part of this document may be copied or reproduced in
any form without prior written consent of MäK Technologies, Inc.

VR-Link, MäK Logger, TLogger and MäK Stealth are trademarks of MäK Technologies, Inc.
Iris Performer and RealityEngine are trademarks of Silicon Graphics, inc.
OpenFlight is a trademark of MultiGen, Inc.
All other trademarks are owned by their respective companies.

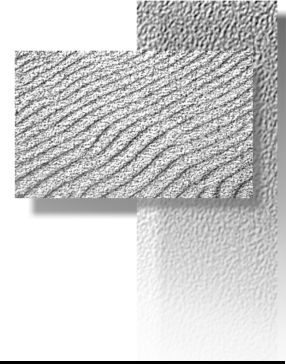
This manual was written and illustrated by Michael Maffie
with technical and editorial assistance from
Len Granowetter, Chris Winters and David LeCompte.

MäK Technologies
185 Alewife Brook Parkway
Cambridge, MA 02138 USA

Voice: 617-876-8085
Fax: 617-876-9208

info@mak.com
www.mak.com

Revision LGSG/W-970611a



Contents

Chapter 1 Introduction

Welcome to the Logger	1-1
1.1. About DIS	1-2
1.2. Other MäK Products for the Virtual World	1-3
1.3. About this Guide	1-4
1.4. Document Conventions	1-5

Chapter 2 Installation

Chapter Overview	2-1
2.1. System Requirements	2-2
2.2. Installing the Logger	2-3
2.3. Obtaining and Installing Upgrades	2-5
2.4. Obtaining Technical Support	2-5
2.5. Contacting MäK	2-6

Chapter 3 General Logger Concepts

Chapter Overview	3-1
3.1. Logger Overview	3-2
3.2. Startup	3-4
3.3. Recording and Playback	3-4
3.4. Playback Timescale and Direction	3-6
3.5. Compensation for Timescale Effects	3-7
3.6. End-of-File Action	3-9
3.7. Network Communications	3-10

Chapter 4 The Graphical Interface

Chapter Overview	4-1
4.1. The Logger GUI Controls	4-2
4.2. Recording with the Logger GUI	4-4
4.3. Playback with the Logger GUI	4-5
4.4. Navigating Time in the Logger GUI	4-6
4.5. Setting Timescale and Direction in the Logger GUI	4-7
4.6. Setting Configuration Options in the GUI	4-9
4.7. Summary of Startup Options for the Logger	4-10

Chapter 5 The TLogger Interface

Chapter Overview	5-1
5.1. Starting the TLogger	5-2
5.2. Recording with the TLogger	5-2
5.3. Playback with the TLogger	5-4
5.4. Navigating Time in the TLogger	5-6
5.5. Controlling Entity Maintenance in the TLogger	5-7
5.6. Setting Playback Timescale and Direction	5-8
5.7. Controlling Heartbeat and Timeout in the TLogger	5-9
5.8. Using a Script to Control Recording and Playback	5-10
5.9. Summary of TLogger Startup Options	5-11
5.10. Summary of TLogger Run-Time Commands	5-12

Chapter 6 Logger Tools

Chapter Overview	6-1
6.1. Viewing Logger Files with LgrDump	6-2
6.2. Concatenating Files with LgrCat	6-4
6.3. Merging Files with LgrMerge	6-6
6.4. Using LgrOffset to Shift PDUs in Time	6-7
6.5. Using LgrTrim to Extract Part of a File	6-8

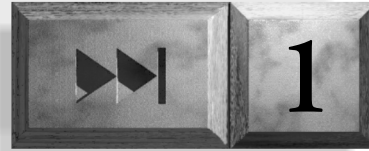
Appendix A Troubleshooting

Chapter Overview	A-1
1.1. Troubleshooting Logger Network Communications	A-2

Appendix B. Logger Manual Pages

lgrTools	B-2
TLogger	B-4
Logger	B-7

Index



Introduction

Welcome to the Logger

The *MäK Logger* is a data recorder that records DIS exercises and plays them back for after-action review. The Logger gives you a choice of a graphical VCR-style interface or a shell-style command line interface.

All versions of the Logger let you scale and reverse time during playback. You can quickly scan a Logger recording to find an area of interest, and then replay that section forward or backward at any speed.

The Logger is available for a range of UNIX platforms as well as for Windows NT and Windows 95. The Windows version also runs under Windows 3.1x/Win32s with certain limitations.

The Logger package includes the following:

- ♦ The **Logger**, a DIS data-recorder/player with a VCR-style graphical user interface.
- ♦ The **TLogger** (or TextLogger), which performs the same functions as the Logger, but has an interactive, character-based shell-style interface. The TLogger adds script support to the Logger's feature set.
- ♦ A set of **Logger Tools** that let you combine, modify and inspect recorded Logger files. You can merge the activity from several exercises into a single Logger file, extract a portion of an exercise to a new file, and perform other file operations.
- ♦ A set of prerecorded sample **Logger recordings** (also called Logger *tapes*).

1.1. About DIS

DIS (Distributed Interactive Simulation) is a means of representing players in a shared virtual world through the exchange of data between applications. These applications allow user interaction and usually run on separate, networked computers.

The DIS protocol is a set of standards that governs how participating applications share information about the virtual world. The protocol specifies a set of packets, called *protocol data units* (PDUs), that communicate this information. Each PDU identifies the sender and receiver, and contains other information depending on the PDU type. The DIS protocol also specifies when and how frequently PDUs are sent.

This release of the Logger includes two sets of executables: one to support DIS 2.0.4 and IEEE version 1278.1-1995, and the other for use with DIS version 2.0.3.

1.1.1. Getting DIS Standards Information

For documentation of the IEEE 1278.1-1995 version of DIS, obtain the publication, *1278.1-1995 IEEE Standard for Distributed Interactive Simulation, Application Protocols*. For more information, visit the IEEE World Wide Web site at:

<http://standards.ieee.org/catalog/index.html>

Search on the string 1278 to find DIS-related material.

To phone IEEE, call 800-678-IEEE (from US and Canada) or 908-981-1393.

You can get an order form for DIS documentation from the DMSTTIAC Web site at:

<http://dmsttiac.hq.iitri.com>

Navigate to the Orlando Service Center, then to the DMSTTIAC catalog. The order form is document number std1001.

1.2. Other MäK Products for the Virtual World

The Logger is a member of MäK Technologies' line of DIS software products. These products are designed to streamline the process of developing and using networked simulated environments.

In addition to the Logger, the MäK DIS line includes:

- ♦ **The VR-Link Network Toolkit.** The Toolkit is an object oriented library of C++ functions and definitions that implement the DIS protocol. This library minimizes the time and effort required to build and maintain new DIS-compliant applications, and to integrate DIS compliance into existing applications.

The Toolkit includes a set of sample debugging applications and their source code. The source code serves as an example of how to use the VR-Link Toolkit to write DIS applications. The executables provide valuable debugging services such as generating a predictable stream of PDUs, and displaying the contents of PDUs transmitted on the network.

- ♦ **The MäK Stealth.** The *Stealth* displays a realistic, three-dimensional view into your virtual world. You can view this world from the inside of a moving DIS vehicle, or place the eyepoint at another moving or stationary location. The Stealth lets you switch rapidly among several predefined viewpoints while the simulation is underway.

- ♦ **Dial-a-Tank.** *Dial-a-Tank* is a reconfigurable simulator program that lets you create and simulate DIS vehicles quickly, without programming. Dial-a-Tank's graphical interface lets you assemble vehicles by choosing from a collection of predefined components. You can adjust the physical properties and positioning of these components to meet your exact needs.

Once your vehicle is complete, you can drive it in a networked simulation. Dial-a-Tank provides a set of GUI controls, and supports PC-compatible steering wheel assemblies and other hardware controls.

1.3. About this Guide

This guide assumes that you are familiar with basic UNIX or DOS operations and that you already know how to work in your computer environment, either OSF/Motif or Windows.

The Logger is a dynamic product, constantly evolving to meet user needs. Consequently, you may find improvements or new features that have been added to the Logger since this manual went to press.

The chapters are organized as follows:

Chapter 1, *Introduction*, is this chapter.

Chapter 2, *Installation*, explains how to install or upgrade the software, and how to obtain technical support.

Chapter 3, *General Logger Concepts*, provides a general overview of what the Logger does, and describes features common to both Logger interfaces. This chapter also explains advanced processing features which may not otherwise be obvious during Logger operation.

Chapter 4, *The Graphical Interface*, shows how to record and play back DIS network activity using the GUI versions of the Logger.

Chapter 5, *The TLogger Interface*, shows how to use the TLogger, the text-based shell-style version of the Logger.

Chapter 6, *Logger Tools*, describes the utility programs that combine, modify and inspect existing Logger recordings.

Appendix A, *Troubleshooting*, shows how to correct network problems that may prevent the Logger from communicating properly.

Appendix B, *Logger Manual Pages*, contains a copy of the Logger's on-line man pages.

1.4. Document Conventions

This document uses the following typographic conventions:

Monospaced Type indicates commands or values you enter, either at the operating system command prompt, the Logger prompt, or in a dialog box.

Monospaced Italic is used in command examples to indicate generic arguments which you replace with appropriate values.

Monospaced Bold is used for illustrations of on-screen text.

[] Brackets are used to indicate optional arguments.

| The pipe symbol separates options in a command where only one option may be chosen at a time.

Bold is used for the names of utilities and other programs.

Italic indicates an object or option that you can select in a graphical interface, such as a button or an option from a scrollable list. Italics are also used for emphasis.

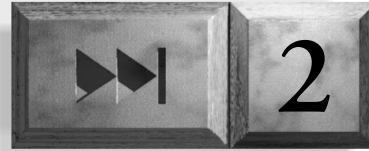
Menu→Option Menu options appear as italicized commands linked together by an arrow. For example, an instruction to select the *Save* option from the *File* menu appears as:
Choose *File→Save*

Directory names are preceded with dot and slant characters that show their position with respect to the VR-Link home directory. For example, the directory:

`vrlink2.4.5/bin3`

appears in the text as:

./bin3



Installation

Chapter Overview

This chapter covers the following topics:

- ♦ System requirements and supported platforms (page 2-2)
- ♦ Installing the Logger (2-3)
- ♦ Obtaining and installing an upgrade (2-5)
- ♦ Technical support (2-5)
- ♦ How to contact MäK Technologies (2-6)

2.1. System Requirements

2.1.1. Hardware Required for Installation

For UNIX platforms, you can install the Logger from a CD-ROM drive, or from a 4mm or ¼-inch tape drive.

The Windows version of the Logger is available only on CD-ROM only, and requires a 486, Pentium or Pentium Pro-based PC.

2.1.2. Disk Space Requirements

The Logger requires 2.5 Mb of hard disk space, plus an additional 33Mb for the sample Logger recordings. If you are installing from a CD and choose not to install the samples during the initial installation, you can copy them later or play them directly from the CD.

2.1.3. Supported Platforms

The Logger is available for an expanding range of UNIX platforms including the following:

- ♦ Silicon Graphics workstations with Irix6.x, Irix5.x or Irix4.x
- ♦ Sun Sparcstation with SunOS 4.x or Solaris 2.4 (or later)
- ♦ DEC Alpha with OSF/1 v3.0
- ♦ HP 9000 Series 700 with HPUX operating system version 9

The following Windows versions are also supported:

- ♦ Windows NT or Windows 95 with the TCP/IP protocol installed.
- ♦ Windows 3.1 or 3.11 with the Win32s extensions and TCP/IP installed. In this environment, only the GUI version of the Logger is supported, and multiple applications cannot share the same UDP port.

Your platform may be supported, even if it does not appear above. Please contact MäK Technologies for an updated list.

2.2. Installing the Logger

UNIX Installation from CD-ROM

1. Mount the CD-ROM and run:
`./install`
2. Follow the on-screen instructions.

You can install the Logger into any directory for which you have write permissions.

UNIX Installation from Tape

To install the Logger from a 4mm tape, create a destination directory, change to that directory, and execute this command:

```
tar xvf tape-device
```

You can also use the above command to install from a ¼-inch tape on all systems except SGI. If you are installing onto an SGI from a ¼-inch tape, execute:

```
tar xvf /dev/tape-devicens
```

Windows Installation from CD-ROM

1. From Windows File Manager or Explorer, run the **Setup.exe** program located in the root directory of the CD-ROM.
2. Follow the on-screen instructions. You will be prompted to confirm the directory where the Logger is to be installed. If desired, you can replace the suggested default with a different destination.

Optional Sample Data Files

If you do not install the sample Logger tapes when prompted during setup, you can copy them to the disk later or play them directly from the CD.

2.2.1. What Gets Installed

The Logger installation process creates a directory structure like the one shown in Figure 2-1. If you have other VR-Link 2.4.5 products installed on your system, you can install the Logger files into your existing VR-Link tree.

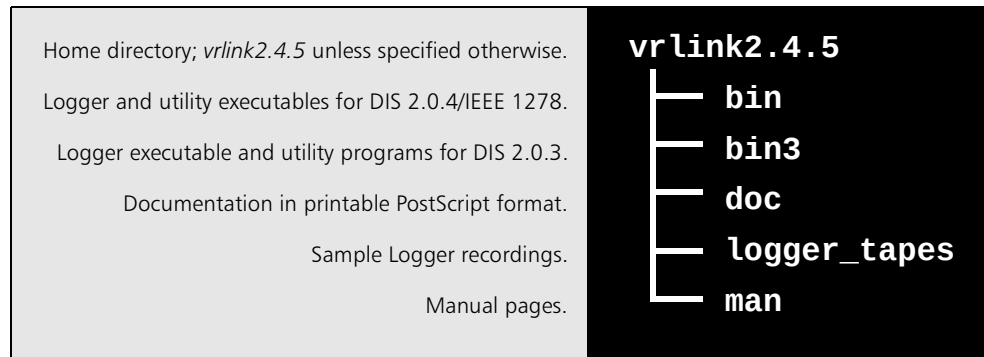


Figure 2-1. Destination of Logger files

The Binary Directories

This package contains two sets of executables. Those for use with DIS version 2.0.4 and IEEE 1278 are in the `./bin` directory; those for version 2.0.3 are in `./bin3`.

The Doc and Man Directories

The `./doc` directory contains the Logger documentation in a printable PostScript format. On UNIX platforms, the `./man` directory contains the Logger manual pages; on the Windows platform, the `.\man` directory contains the manual pages as plain text files.

Logger Tapes

The `logger_tapes` directory contains recordings of two DIS exercises. One exercise is provided in both DIS 2.0.4 and 2.0.3 versions. The other is provided in a DIS 2.0.4 format only. For more information about these recordings, please see [section 3.1.4](#), “*Sample Logger Tapes*”.

2.3. Obtaining and Installing Upgrades

If you would like to receive upgrades, please contact the MäK Technologies sales department for information about how to establish a VR-Link Maintenance Agreement. MäK ships upgrades to qualified users on CD-ROM or tape, or provides instructions for downloading the most recent release via anonymous FTP.

If you receive an upgrade on tape or CD-ROM, install it following the procedures in [section 2.2, “Installing the Logger”](#).

To install a UNIX upgrade downloaded via FTP, uncompress and untar it as follows:

1. Move the downloaded file to the Logger home directory. Uncompress the file with the following command:

```
uncompress filename.tar.Z
```

2. Extract the release from the resulting tar file with this command:

```
tar xvf filename.tar
```

2.4. Obtaining Technical Support

MäK Technologies’ technical support welcomes your questions and comments about the Logger. If you need to resolve a problem, please include the following information in your communication:

- ♦ The version of the Logger you are using.
- ♦ A description of the problem.
- ♦ Any relevant information about your host configuration, including your operating system, compiler version, network configuration, and any other information that may be related to the problem.

For technical support addresses and other contact information, please see [section 2.5](#).

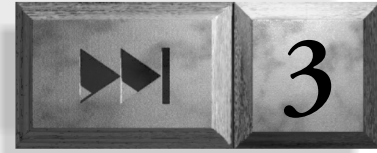
2.5. Contacting MäK

For Logger technical support and information about upgrades and other MäK products, you can reach us by e-mail, Web, postal mail, fax, or voice phone:

For sales and upgrade information by e-mail:	<code>vrlink-info-sales@mak.com</code>
For technical support by e-mail:	<code>vrlink-tech-spt@mak.com</code>
For general MäK information via the Web:	http://www.mak.com

Send postal correspondence to:	MäK Technologies 185 Alewife Brook Parkway Cambridge, MA 02138
--------------------------------	--

Call or fax us at:	Fax: 617-876-9208
	Voice Phone: 617-876-8085



General Logger Concepts

Chapter Overview

This chapter provides a general overview of what the Logger does, and describes features common to both Logger interfaces. For step-by-step instructions about how to operate the Logger, see chapters [4](#) and [5](#).

This chapter covers the following topics:

- ♦ A general description of the Logger and its interfaces (page [3-2](#))
- ♦ DIS versions compatible with the Logger([3-3](#))
- ♦ Differences between UNIX and Windows versions ([3-2](#))
- ♦ Sample Logger recordings ([3-3](#))
- ♦ What happens during recording and playback ([3-4](#))
- ♦ Timescale and direction during playback ([3-6](#))
- ♦ How the Logger compensates for timescale effects ([3-7](#))
- ♦ Controlling the Logger's end-of-file action ([3-9](#))
- ♦ Specifying the network addresses used by the Logger for sending and receiving data ([3-10](#))

3.1. Logger Overview

The Logger is a data recorder that records DIS exercises and plays them back for after-action review. In record mode, the Logger monitors all DIS traffic on a specified UDP port, and stores it in a file for future playback. In playback mode, the Logger transmits DIS data packets stored in an existing Logger recording.

3.1.1. Logger Interfaces

The Logger provides two interfaces: a graphical, VCR-style interface (GUI), and a character-based command-line interface (CUI). Although these interfaces have a different look and feel, they have nearly identical capabilities. Both let you:

- ♦ Start, stop or pause when playing a recording.
- ♦ Navigate to a specific time-point when playing a recording.
- ♦ Control playback speed and direction.
- ♦ Select the network addresses used for the sending and receiving of data.

The only important functional differences between the graphical and character-based Loggers are:

- ♦ You can use a script to control the CUI-based version (the *TLogger*).
- ♦ The GUI lets you change various options while the program is running, while the CUI requires that you set these options at startup through command-line switches.

To start the Logger with the graphical interface, you can run **Logger** from the UNIX console (or a DOS box within Windows), or launch the Logger from its Windows icon. To use the text-based interface, run **TLogger**.

3.1.2. Differences Between UNIX and Windows Versions

Although the UNIX and Windows versions of the Logger operate in almost exactly the same way, there are a few minor differences.

On UNIX systems, you can use the *man* utility to view the Logger's on-screen manual pages. The Windows version includes these pages as plain text files, located in the `.\man` directory. To view them, use the DOS **type** command or another text viewer or editor. You can also view the man pages in the Logger's online Windows help system.

Under UNIX, the entire Logger command line is case-sensitive, while under Windows, only the option letters are case-sensitive. For example, the `-p` option is different from the `-P` option, regardless of environment.

For consistency across platforms, the hyphen is always used to precede command-line arguments, rather than the slash character typically used with DOS command-line arguments.

3.1.3. Supported DIS Revisions

This release of the Logger supports version 2.0.3, 2.0.4, and the IEEE 1278.1-1995 version of the DIS Protocol. For information about DIS revisions, see section 1.1.1, *“Getting DIS Standards Information”*.

Logger Executables for Different DIS Versions

This release contains two sets of executables for use with different DIS versions. The `./bin` directory contains Logger executables and utility programs that are compatible with DIS 2.0.4 and IEEE 1278.1. The `./bin3` directory contains executables for use with DIS version 2.0.3.

3.1.4. Sample Logger Tapes

The Logger release includes sample recordings of two DIS exercises:

- ♦ An exercise from the December 2, 1993 I/ITSEC show, Orlando, Florida. Two versions of this exercise are included with the Logger:

`iitsec93.tape2` — contains DIS 2.0.3 data
`iitsec93.dis204` — contains DIS 2.0.4 data.

This exercise has been included with previous Logger releases. To view the exercise with the MäK Stealth, run the *run-demo-low* script included with the Stealth.

- ♦ An exercise prepared for the MäK/MultiGen/SGI demonstration at the 1995 I/ITSEC show:

`montyDemo.lgr` — contains DIS 2.0.4 data.

This recording is of a DIS 2.0.4 exercise in which two A-10s attempt to destroy seven ground targets near an airfield.

To view this exercise with the MäK Stealth, run the *run-demo-hi* script included with the Stealth. Unlike most Logger recordings, this demo contains *View Control PDU*s which cause the Stealth to follow the action, switching among many predetermined viewpoints.

3.2. Startup

To start the TLogger, you must specify whether it opens in record or playback mode, and specify the name of the file to open for recording or playback. Use the `-r` switch to open a file for recording, or the `-p` switch to open a file for playback:

```
TLogger -r newfile.lgr
TLogger -p demofile.lgr
```

The `-p` and `-r` switches are optional for the GUI Logger; if you don't use a switch, you can use options on the *File* menu to select the operating mode and data file. To start the GUI Logger, run:

Logger

from the UNIX console (or DOS box), or use the Logger's Windows icon.

3.3. Recording and Playback

3.3.1. What Happens During Recording

In record mode, the Logger records all DIS packets on UDP port 3000, without regard to exercise ID. You can specify a different UDP port at startup by using the `-P` switch, for example:

```
Logger -r newfile.lgr -P 2727
```

With each PDU recorded, the Logger stores a timestamp indicating when the PDU occurred relative to the start of the recording.

When operating in record mode, the Logger is in either a *recording* state or a *stopped* state. When stopped, the Logger does not record data and does not advance time. When running the TLogger, you can select a recording or stopped state with the `go` and `stop` run-time commands. In the GUI Logger, use the *Record* and *Stop* VCR buttons to control the state.

3.3.2. What Happens During Playback

When you start the Logger and specify a file for playback, the Logger sends the recorded PDUs to UDP port 3000. You can use the `-P` switch to specify a different port, for example:

```
Logger -p demofile.lgr -P 2727
```

The Logger may or may not begin playback immediately on startup, depending on which additional command-line switches you use.

When operating in playback mode, the Logger is in either a *playing* state, a *paused* state or a *stopped* state. When playing, the Logger reads DIS packets from the recorded file and transmits them in sequence over the network. When paused, the Logger halts at the current timepoint in the file and repeatedly retransmits PDUs that represent the current state of each entity. When stopped, the Logger halts at the current timepoint in the file and stops sending data.

In the TLogger, you can select a playing, paused, or stopped state with the `go`, `pause`, and `stop` commands. In the GUI Logger, use the *Play*, *Pause* and *Stop* VCR-buttons to control the state.

During playback, you can control the playback speed and direction to produce fast motion, slow-motion and reverse playback effects. Sections [3.4](#) and [3.5](#) describe these time scaling features in detail.

Both Logger versions let you navigate to a specified timepoint in a recording, or skip forward or backward by a specified time increment. In the GUI, you navigate time using the *Start* and *Stop* buttons, and the *Skip Time* and *Set Time* menu options (described in section [4.4](#)). In the TLogger, you use the `start`, `end`, `time` and `skip` commands (see section [5.4](#)).

3.4. Playback Timescale and Direction

The Logger lets you play a recording at any speed within the capabilities of your platform, forward or backward.

You determine the speed by specifying a *timescale* value, the factor by which time is multiplied during playback. The default timescale setting of 1.0 causes a Logger file to play forward at normal speed. A timescale of 0.5 results in forward, half-speed playback. A negative timescale causes reverse playback. Figure 3-1 shows a set of typical timescale values their effect on speed and direction.

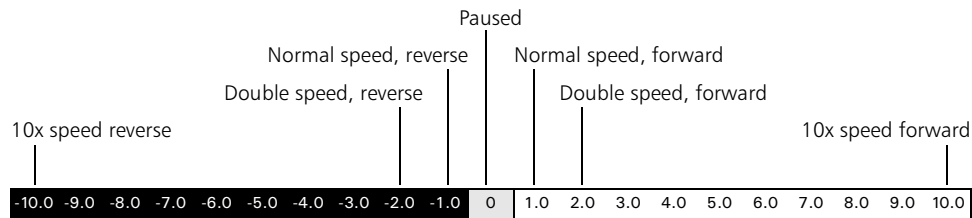


Figure 3-1. Sample Timescale Values

In the GUI Logger, you use a slider control to set the timescale or enter the timescale value as a numeric value (see section 4.5). In the TLogger, you control timescale with the `scale` command (see section 5.3).

You can specify an initial timescale at Logger startup with the `-S` command-line option, for example:

```
Logger -p my-exercise.lgr -s 2.5
```

3.5. Compensation for Timescale Effects

The Logger has corrective features that adjust PDU frequency and content for non-normal playback speeds. *Interim PDU generation* prevents receiving applications from inappropriately timing out entities during slow-speed playback. *Velocity and acceleration scaling* prevents anomalies in vehicle movement that would otherwise occur at any non-standard playback speed.

3.5.1. Generating Interim PDUs During Slow-Speed Playback

Heartbeat and Timeout

In a DIS exercise, each participating entity sends PDUs to other applications at a regular time interval, called the *heartbeat*. A common heartbeat rate is five seconds.

Most applications that receive PDUs expect to receive them from each entity at the agreed-upon heartbeat rate. If a receiving application does not receive a PDU from an entity for an unusually long time, it may conclude that the entity no longer exists, and then time the entity out. A typical *timeout* threshold is 12 seconds, or 2.4 times the heartbeat rate.

Using Interim PDUs to Maintain Heartbeat

During slow-speed playback, the long time-lapse between an entity's recorded PDUs might cause receiving applications to time the entity out prematurely. To prevent this, the Logger generates interim PDUs as needed to maintain each entity's heartbeat.

The Logger calculates the contents of an interim PDU by applying a dead-reckoning algorithm to an entity's most recent PDU.

By default, the Logger ensures that at least one PDU is transmitted for each entity every five seconds. You can specify a different heartbeat rate. In the GUI, the heartbeat setting is in the *Configuration* dialog box, available through the *Options* menu (see section 4.6). In the TLogger, use the `heartbeat` run-time command (see section 5.7).

Scaling the Timeout Period During Slow-Speed Playback

When an entity's PDUs stop occurring in the recording, the Logger continues to generate interim PDUs for the entity for the specified *timeout* period.

During slow-speed playback, the timeout period is extended accordingly. For example, when playing a recording at half-speed, using the default timeout setting of 12 seconds, the Logger will generate interim PDUs for an entity for 12 *recorded* seconds, or 24 real-time seconds before timing it out.

To set a different timeout threshold in the GUI, change the timeout setting in the *Configuration* dialog box, available through the *Options* menu. In the TLogger, use the `timeout` run-time command. See sections 4.6 and 5.7 for details.

Updating Entity Information Quickly After a Time Jump

During playback, the Logger lets you jump to a specified timepoint using the *Set Time* or *Skip Time* option in the GUI, or the `time` or `skip` command in the TLogger. After the time jump, the Logger resumes sending PDUs for all entities, starting at the new timepoint. At normal-speed playback, the Logger reads and sends updated PDUs for most entities within five seconds after the time jump (or within the heartbeat rate used when the recording was made).

However, during slow-speed playback, it may take much longer for the Logger to send updated PDUs for all entities. For example, if you jump to a new timepoint when playing a recording at one-tenth speed, fifty seconds may pass before the Logger has sent a new PDU for every entity. During this time, receiving applications will reflect entity states that existed before the time jump, or may time entities out.

To prevent these anomalies, the Logger provides a feature called *entity maintenance*. After a time jump, the Logger can generate up-to-date interim PDUs to fill the period before the transmission of each entity's first recorded PDU. To create the interim PDUs, the Logger applies a dead reckoning algorithm to each entity's most recent recorded PDU *prior* to the new time point.

In the GUI Logger, entity maintenance is applied when you issue a *Skip Time* command, but not for a *Set Time* command.

In the TLogger, you can turn entity maintenance on or off separately for the `time` and `skip` commands with the `timemaint` and `skipmaint` commands. See section 5.5 for details.

Because additional processing is required by entity maintenance, you will experience a delay when jumping to a new timepoint. The delay length depends on the number of entities processed and the size of the time jump.

3.5.2. Velocity and Acceleration Scaling

When playing at speeds above or below normal, the Logger can adjust the velocity and acceleration values in entity state PDUs accordingly. This scaling allows the receiving DIS applications to perform accurate dead reckoning.

For example, if the Logger is playing at double speed, it can double the values in the PDU velocity and acceleration fields. This adjustment causes a vehicle to be move at double its original speed when viewed on a Stealth. Without this scaling, a moving vehicle would appear to be moving at its original slower speed, and would have to lurch forward with the arrival of each new PDU in order to catch up with the new location data contained in the PDU.

This scaling feature, also called RVA or Remote Vehicle Approximation, is active by default. To disable it, use the `-d` command-line option at Logger startup. Note that RVA must be enabled in order for the Logger to send PDUs while in the paused state.

Note: If a file contains bundled PDUs, the Logger does not unbundle them on playback, and therefore cannot scale their speed and velocity. If this is a problem, contact MäK for information about the `lgrUnbundle` tool.

3.5.3. Effect of Reverse Playback on Dead Reckoning

Applications receive PDUs in reverse order during reverse playback. Because dead-reckoning is based on acceleration, and acceleration is not a mathematically reversible operation, your applications may place a vehicle on a path different from the one originally traveled.

3.6. End-of-File Action

When the Logger reaches the end of a file, it can enter a stopped or paused state, or it can wrap to the other end of the file and continue playback. To specify an end-of-file action, start the Logger with the `-e` command-line option:

Logger	<code>-es</code>	Stop (default)
Logger	<code>-ep</code>	Pause
Logger	<code>-ew</code>	Wrap

In the Logger GUI, you can also change the end-of-file action in the *Configuration* dialog box, available through the *Options* menu.

During reverse playback, if *wrap* is in effect and the beginning of the tape is reached, the tape wraps to the end and continues reverse playback.

3.7. Network Communications

The Logger can direct its output to a specific destination address during playback, and can listen (subscribe) to one or more multicast addresses during recording. You can also control which UDP port is used for playback or recording.

3.7.1. Specifying a Destination Address for Outgoing Data

Normally, the Logger sends outgoing data packets to the host machine's default broadcast address. To specify a different destination address, use the `-A` command-line option, as in this example:

```
Logger -p my_exercise.lgr -A 123.234.123.234
```

The specified destination address may be a broadcast, multicast, or unicast address.

3.7.2. Subscribing to Multicast Groups

In record mode, the Logger can subscribe to multicast groups through use of the `-S` command-line option. When subscribed to multicast groups, the Logger also continues to receive data through the machine's broadcast and unicast addresses.

To subscribe to more than one multicast address, use the `-S` option several times on the same command line:

```
Logger -r newfile.lgr -S 225.0.0.118 -S 225.0.0.202
```

Subscription pertains to record mode only.

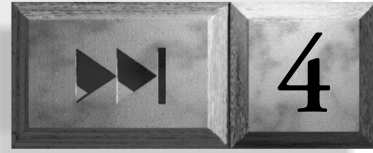
Multicasting is fully supported under IRIX-5, IRIX-4, and Solaris-2. Under DEC OSF/1 v2.0, applications can receive, but not send, multicast.

Note: Multicast subscription is a property of the host machine, not of individual applications. Therefore, if several applications are running on the same machine using the same port, and those applications subscribe to multicast groups, all the applications will receive packets for all multicast groups to which any application is subscribed. Accordingly, if the Logger is running on the same machine and using the same port as other applications that subscribe to multicast groups, the Logger may receive traffic intended for the other applications.

3.7.3. Specifying a UDP Port

By default, the Logger sends and receives data over UDP port 3000. You can specify a different port by using the uppercase `-P` switch, as in these examples:

```
Logger -P 2727
Logger -p mydemo.lgr -P 2727
```



The Graphical Interface

Chapter Overview

This chapter covers the following topics as they apply to the Logger's graphical interface:

- ♦ Starting the GUI Logger (page [4-3](#))
- ♦ Making a recording ([4-4](#))
- ♦ Playing back a recorded file ([4-5](#))
- ♦ Navigating time ([4-6](#))
- ♦ Controlling the scale and direction of time during playback ([4-7](#))
- ♦ Setting configuration options ([4-9](#))
- ♦ Summary table of command-line options for the GUI Logger ([4-10](#))

4.1. The Logger GUI Controls

The Logger GUI provides a separate control set for playback and record modes. In playback mode, the GUI provides a slider for scaling and reversing time, and a set of time navigation controls. In record mode, the Logger presents a simplified interface with recording controls only.

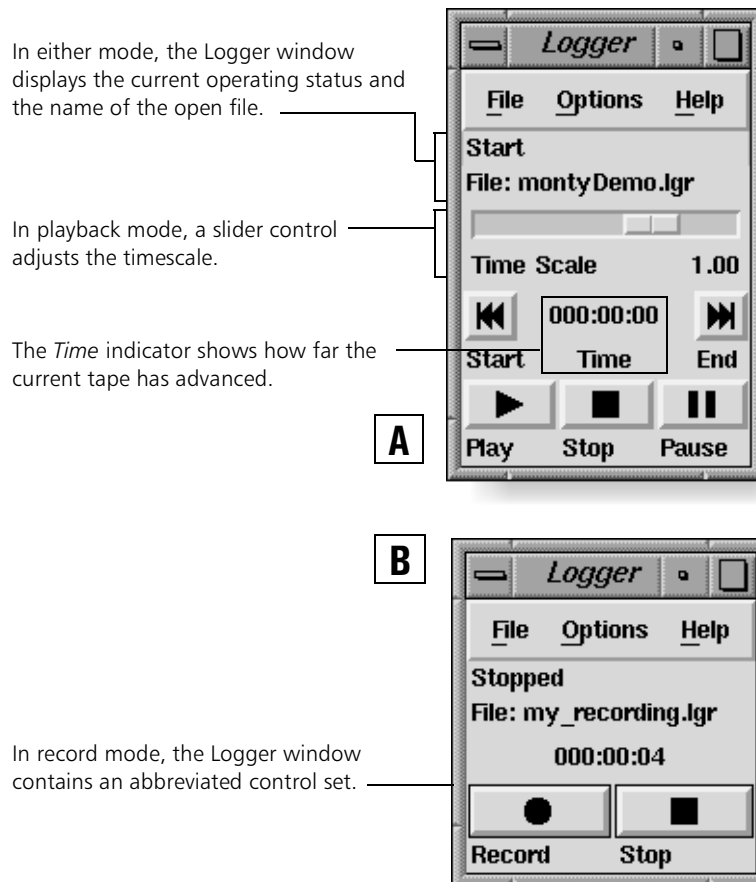


Figure 4-1. The Logger GUI in playback mode (A) and record mode (B).

4.1.1. Starting the Logger GUI

You can use command-line options to set many of the Logger's initial settings. However, because the GUI provides run-time access to most frequently-used options, you can usually start the Logger without options.

A complete list of startup options for GUI operation appears on page [4-10](#).

Logger Startup on UNIX

To start the Logger GUI, open a UNIX shell, switch to the /bin or /bin3 directory, and enter:

```
Logger [options]
```

Logger Startup in Windows

Under Windows, you can start Logger GUI from a DOS prompt or from an icon within the Windows GUI. To start Logger from the DOS prompt, open a DOS box, switch to the \bin or \bin3 directory, and enter:

```
Logger [options]
```

When the Logger window and its companion console window appear, you can close the initial DOS box if desired.

To run the Logger from the Windows GUI, double-click the Logger icon in Program Manager (Windows NT 3.5 and Windows 3.1x), or choose the icon from the Start Menu or folder (Windows NT 4.x and Windows 95).

Note: Some options (such as DIS port or destination address) are not available through the Logger GUI, and must be specified on the command line at startup.

To invoke command-line options when starting the Logger from an icon, add the desired options to the startup command line in the icon's property sheet. You can create several Logger shortcuts (or icons, in Windows 3.1), each with a different set of command-line options. See your Windows documentation for more information.

4.2. Recording with the Logger GUI

To begin recording, choose **File** → **Record new file**. When the **Select File** dialog box appears, enter a new filename or select an existing file to overwrite. Click **OK** to close the dialog box.

If necessary, use the
 Directories window to
 navigate to the desired
 directory. _____

Select an existing file to
 overwrite from the Filelist,
 or enter a new filename in
 the Selectionbox. _____

Figure 4-2. Specifying a destination file for recording.

After you select the file, the Logger enters record mode in a stopped state. To begin recording or return to a stopped state, click on the appropriate VCR-button. The Status field displays either **Recording** or **Stopped** accordingly.

As recording progresses, the Time field is updated to display elapsed time. In record mode, the timescale is fixed at 1.0 (normal speed).

The Logger records all DIS traffic that the host machine receives on the default UDP port (3000). If desired, you can use startup command-line options to specify a different UDP port or subscribe to additional multicast addresses. For more information about these options, see [section 3.7; "Network Communications"](#)

4.3. Playback with the Logger GUI

To play back an existing file, choose *File→Play back old file*. When the *Select File* dialog box appears, navigate to the desired directory, select the file to be played, then click *OK*.

By default, the Logger enters a stopped state when the file is loaded. To play, pause or stop playback, click on the appropriate VCR-button.

If the recording is at its beginning or end, the status field displays *Start* or *End*. Otherwise, *Stopped*, *Paused*, or *Playing* is displayed to reflect the Logger's current operating state.

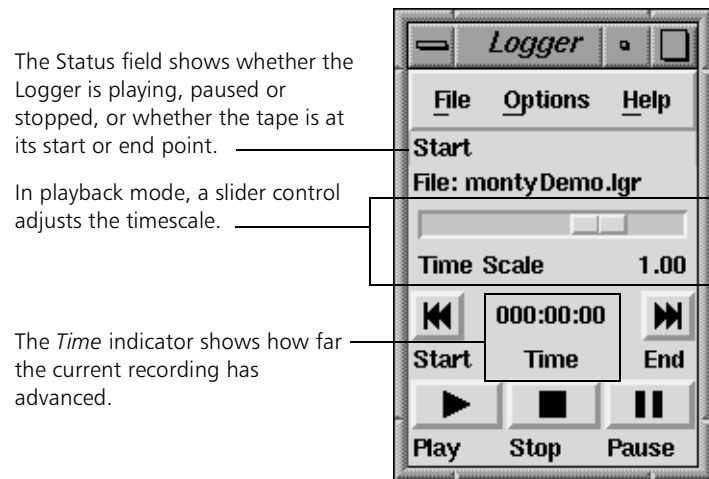


Figure 4-3. Logger GUI playback controls.

The Logger sends data to host machine's default broadcast address over the default UDP port (3000). If desired, you can use command-line options to specify a different UDP port or destination IP address. See [section 3.7, "Network Communications"](#) for more information. The Logger transmits PDUs while in the playing or paused states. In the stopped state, no PDUs are sent.

4.4. Navigating Time in the Logger GUI

The Logger GUI provides three ways to navigate to a specific timepoint:

- ♦ Use the *Start* and *End* buttons in the Logger's main window to jump to the beginning or end of the recording.
- ♦ Use the *Skip Time* option to jump forward or backward in time by a specified offset.
- ♦ Use the *Set Time* option to jump to a specific timepoint.

The *Skip Time* and *Set Time* options are available on the *Options* menu.

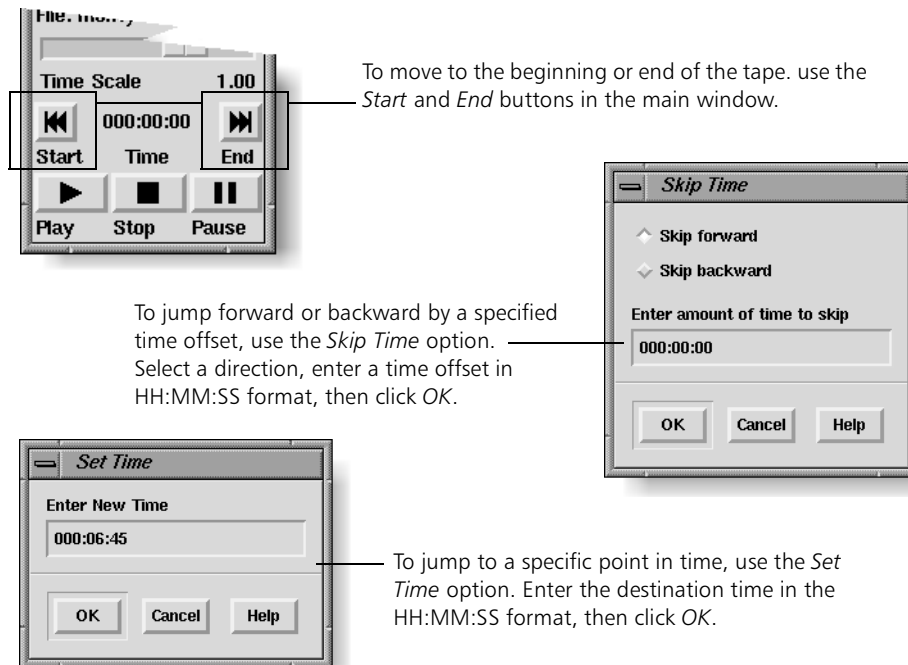


Figure 4-4. Time Navigation in the Logger GUI.

If you execute a time jump while playback is paused or stopped, the Logger remains in that state. The Logger also retains a play state after a time jump unless the end of the file is reached and wrapping is disabled. For more information, see [section 3.6, “End-of-File Action”](#), and [section 4.6, “Setting Configuration Options in the GUI”](#).

During slow-speed playback, when you use the *Skip Time* option, the Logger performs *entity maintenance* to speed the transmission of up-to-date PDUs. When you use the *Set Time* option, the Logger does not perform entity maintenance. For more information, see [“Updating Entity Information Quickly After a Time Jump”](#) under [section 3.5.1](#).

4.5. Setting Timescale and Direction in the Logger GUI

The Logger lets you play a recording forward or backward at any speed, limited only by the capabilities of your hardware. To set the timescale, you can:

- ♦ Use a slider control in the Logger main window. The slider can be customized to contain a specific set of timescale values.
- ♦ Specify an initial timescale at Logger startup through a command-line option.
- ♦ Enter the desired timescale into a dialog box.

For information about choosing timescale values, see [section 3.4, “Playback Timescale and Direction”](#).

4.5.1. Using and Customizing the Timescale Slider

The Timescale slider control in the Logger main window is the most convenient way to change playback speed and direction. In its default configuration, the slider contains nineteen preset values ranging from -10 to +10.

To change the timescale, just drag the slider button. If you need finer control, click once on the slider button to give it focus, then use the left and right arrows on your keyboard to step through the values.

You can determine which values that appear on the timescale slider. At Logger startup, use the `-t` command-line option followed by a comma-separated list of values, as shown below. Do not include any spaces in the value list:

```
Logger -t-5, -4, -3, -2.5, -2, -1.5, -1, 0, 1, 1.5, 2, 2.5, 5, 7
```

You can enter the values in any sequence; they will appear on the slider in ascending order.

4.5.2. Setting an Initial Timescale at Startup

To specify an initial timescale at Logger startup, use the `-S` command-line option, as in this example:

```
Logger -p my-exercise.lgr -s 2.5
```

4.5.3. Entering a Specific Timescale Value During Playback

If the timescale you want is not available through the slider control, you can specify it directly. Choose *Options*→*Set Timescale*, enter the desired timescale into the dialog box, then click *OK*.

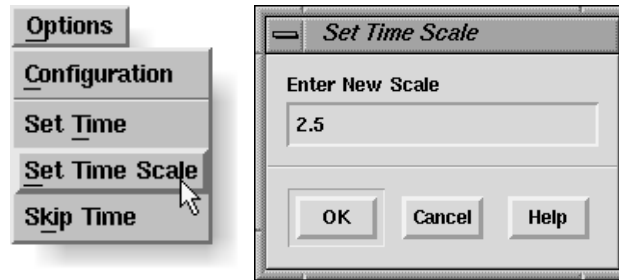


Figure 4-5. The Set Timescale Dialog Box.

If desired, you can enter the timescale value in scientific notation, for example, **1e+4** instead of **10,000**.

Note: Although the Logger lets you specify a very large timescale (beyond 1e+35), the use of an unusually large timescale may produce unexpected results because of hardware limitations.

If a recording is played at a very high speed, the Logger may be unable to keep up, especially during passages that contain a high concentration of data. If the Logger falls significantly behind, it displays the *Timescale* value in red and continues to process the PDUs as quickly as possible. No PDUs are discarded.

4.5.4. PDU Processing for Non-Standard Timescales

When playing at speeds above or below normal, the Logger makes corrective adjustments to the velocity and acceleration values in outgoing entity state PDUs. For details, see [section 3.5.2, “Velocity and Acceleration Scaling”](#).

The Logger can generate interim PDUs during slow-speed playback to prevent receiving applications from timing entities out too soon. For more information, see [section 3.5.1, “Generating Interim PDUs During Slow-Speed Playback”](#), and [section 4.6, “Setting Configuration Options in the GUI”](#).

4.6. Setting Configuration Options in the GUI

To access the *Configuration* dialog box, choose *Options*→*Configuration*.

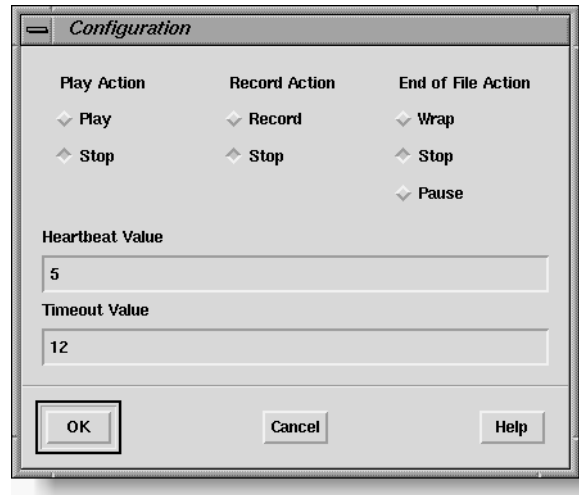


Figure 4-6. The Configuration Dialog Box.

The *Configuration* dialog box contains the following options:

Play Action. This option determines whether the Logger enters a stopped state (default) or begins playing immediately when a file is opened for playback. To set this option at Logger startup, use the `-OS` or `-Op` option.

Record Action. This option determines whether the Logger enters a stopped state (default) or begins recording immediately when a file is opened for recording. To set this option at Logger startup, use the `-bs` or `-br` option.

End of File Action. When the end of a Logger tape is reached during playback, this option determines whether the Logger stops, pauses, or wraps around to the other end of the tape to continue playback (*stop* is the default). To set the end-of-file action at startup, use the `-es`, `-ep`, or `-ew` option. See [section 3.6, “End-of-File Action”](#) for more information.

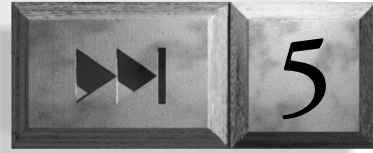
Heartbeat. During slow-speed playback, this option determines how frequently (in seconds) the Logger generates interim PDUs for each entity. For information, see [“Using Interim PDUs to Maintain Heartbeat”](#) under [section 3.5.1](#).

Timeout. During slow-speed playback, this option determines how long the Logger continues to generate interim PDUs for an entity once that entity’s PDUs have stopped occurring in the recording. For more information, see [“Scaling the Timeout Period During Slow-Speed Playback”](#) under [section 3.5.1](#).

4.7. Summary of Startup Options for the Logger

Table 4-1: Command-line options for the GUI Logger.

	Option	Description
Common	-h, -help, -?	Displays a brief summary of Logger command-line options.
	-P <i>dis_port</i>	Specifies which DIS port (UDP port) the Logger uses to send or receive data. The default port is 3000.
Record Mode Only	-b <i>state</i>	Specifies whether the Logger enters a stopped state or begins recording immediately when opening a file for recording. Set <i>state</i> to r for record, or s for stopped (GUI default).
	-r [<i>filename</i>]	Causes the Logger to open in record mode; specifies the destination file for recorded data. With the GUI Logger, the -r switch is optional, and may be used without a filename.
	-S <i>mcastAddr</i>	Subscribes the Logger to the multicast address specified by <i>mcastAddr</i> . To subscribe to several multicast addresses, use this option more than once on the command line. For information, see section 3.7.2, “Subscribing to Multicast Groups” .
Playback Mode Only	-A <i>addr</i>	Sets the default IP address to which the Logger sends PDUs. If this option is not used, the Logger uses the computer’s broadcast address. See section 3.7.1, “Specifying a Destination Address for Outgoing Data” .
	-d	Disables the generation of remote vehicle approximation (RVA) data when playing at speeds above or below normal. See section 3.5.2, “Velocity and Acceleration Scaling” .
	-e <i>action</i>	Specifies the Logger’s action upon reaching the end of the file. Set <i>action</i> to p to pause at the file’s end, w to wrap to the other end of the tape and continue, or s to stop (default).
	-o <i>state</i>	When a file is opened for playback, specifies whether the Logger enters a stopped state, or begins playing immediately. Set <i>state</i> to p for playing, or s for stopped (GUI default).
	-p [<i>filename</i>]	Causes the Logger to open in playback mode; specifies the source file to be played. With the GUI Logger, the -p switch is optional, and may be used without a filename.
	-s <i>timescale</i>	Sets the initial playback timescale factor. For more information, see section 3.4, “Playback Timescale and Direction” .
	-T <i>timepoint</i>	Sets the timepoint in the file where playback begins. Enter the <i>timepoint</i> in HH:MM:SS format. Default is 00:00:00
	-t <i>list</i>	Specifies the list of values that appear on the Logger timescale slider control. Enter <i>list</i> as a series of comma-separated values without spaces. See section 4.5.1, “Using and Customizing the Timescale Slider” .



The TLogger Interface

Chapter Overview

This chapter covers the following topics as they apply to the TLogger and its character-based interface:

- ♦ Making a recording (page [5-2](#))
- ♦ Playing back a recorded file ([5-4](#))
- ♦ Viewing the TLogger's recording or playback status ([5-3](#), [5-5](#))
- ♦ Navigating time during playback ([5-6](#))
- ♦ Controlling the scale and direction of time during playback ([5-8](#))
- ♦ Setting heartbeat rate and timeout threshold for slow-speed playback ([5-9](#))
- ♦ Using a script to control recording or playback ([5-10](#))
- ♦ Summary table of TLogger command-line options ([5-11](#))
- ♦ Summary table of TLogger run-time commands ([5-12](#))

5.1. Starting the TLogger

The TLogger's character-based interface offers the same functionality as the graphical interface, plus script-based recording and playback capability.

When starting the TLogger, the most important differences from GUI startup are:

- ♦ You must specify playback or record mode at startup with the **-p** or **-r** switch, respectively. (With the GUI Logger, these switches are optional.)
- ♦ The **-p** or **-r** switch must be followed by the name of the file to be played or recorded. (With the GUI, the filename is optional, even when **-p** or **-r** is used.)

5.2. Recording with the TLogger

To begin recording, open a UNIX shell or DOS box, then start the Logger with the **-r filename** argument:

```
TLogger -r my-exercise.lgr
```

This command causes the TLogger to begin recording all DIS traffic that the host machine receives on UDP port 3000. If desired, you can use command-line options to specify a different UDP port or to subscribe to multicast groups. See [section 3.7, “Network Communications”](#) for more information.

At startup, a status message shows the current port, broadcast address, destination file and current operating status.

```
recordfile my-exercise.lgr
NetSocket ok for port 3000 to 123.456.789.255
Status: Select File
Status: Recording
Logger file my-exercise.lgr is version 2 (writing)
my-exercise.lgr
Cannot adjust timescale while recording.
Type "i" for information, "?" for help.
The Current Status is : Recording
```

Figure 5-1. TLogger recording startup message.

5.2.1. TLogger Run-time Commands for Recording

Recording with the TLogger is straightforward. At the TLogger prompt, enter:

<code>stop</code>	To stop recording without ending the session
<code>go</code>	To begin or resume recording.
<code>quit</code>	To close the file and exit the TLogger.

5.2.2. Displaying Recording Status

At any time during the recording, you can enter the `info` command (or the letter `i`) to display a brief status summary. The summary shows the destination file name, the time elapsed since the start of recording, and the current operating status.

```
> info
FILE: my-exercise.lgr
TIME: 000:01:43
SCALE: 1.000000
STATE: Recording
>
```

Figure 5-2. TLogger recording status summary.

5.2.3. Beginning a Recording Session in a Stopped State

By default, when you start the TLogger in record mode, it begins recording immediately. To begin in a stopped state, use the `-b` command-line option to set the *record action* as in this example:

```
TLogger -bs -r my-exercise.lgr
```

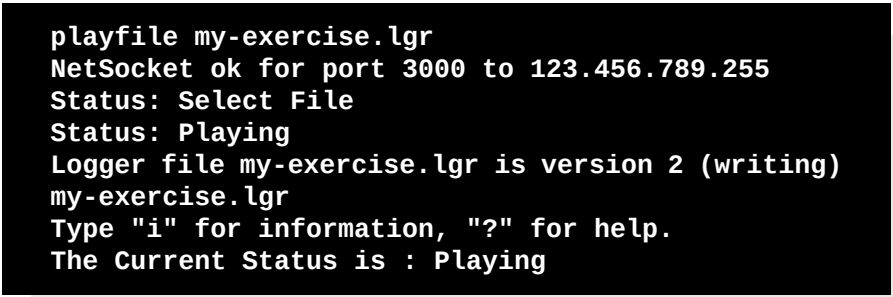
5.3. Playback with the TLogger

To play back a Logger file, open a UNIX shell or DOS box, then start the TLogger with the `-p filename` argument:

```
TLogger -p ../logger_tapes/my-exercise.lgr
```

This command opens the specified file, and sends the recorded DIS information to the host machine's default broadcast address over UDP port 3000. If desired, you can use command-line options to specify a different UDP port or destination IP address. See [section 3.7, "Network Communications"](#) for more information.

A status message shows the current port, destination IP address, source file and current operating status.

A screenshot of a terminal window showing the TLogger playback startup message. The text is as follows:

```
playfile my-exercise.lgr
NetSocket ok for port 3000 to 123.456.789.255
Status: Select File
Status: Playing
Logger file my-exercise.lgr is version 2 (writing)
my-exercise.lgr
Type "i" for information, "?" for help.
The Current Status is : Playing
```

Figure 5-3. TLogger playback startup message in the CUI.

5.3.1. Basic TLogger Playback Commands

In playback mode, the TLogger enters a playing state at startup by default. Thereafter, it is in either a playing, paused or stopped state. At the TLogger prompt, enter:

- | | |
|--------------|--|
| go | To begin or resume playback. When playing, the TLogger sends out PDUs in chronological sequence |
| pause | To pause playback. When paused, the TLogger stops time and repeatedly rebroadcasts each entity's most recent entity state PDU. |
| stop | To stop playback. When stopped, the TLogger stops time and does not transmit data |
| quit | To close the file and exit the TLogger. |

5.3.2. Displaying Playback Status

To display a playback status summary, enter the `info` command (or the letter `i`) at the TLogger prompt. The summary shows the source file name, the current timepoint in the Logger file, the timescale (playback speed), and the current operating status.

```
> info
FILE: my-exercise.lgr
TIME: 000:00:09
SCALE: 1.000000
STATE: Playing
```

Figure 5-4. TLogger playback status summary.

5.3.3. Beginning a Playback Session in a Stopped State

By default, when you start the TLogger in playback mode, it begins playback immediately. To begin in a stopped state, use the `-O` command-line option to set the *play action* as in this example:

```
TLogger -os -p demo.lgr
```

5.4. Navigating Time in the TLogger

During playback, four commands let you navigate to a specific point in a recording:

start	moves to the beginning of the recording.
end	moves to the end of the recording.
time	sets the timepoint to a specified time. The command format is: time HH:MM:SS
skip	causes the Logger to jump forward or backward in time by a specified offset. Enter the offset in an HH:MM:SS format, or as a number of seconds. Use a negative value to skip backward in time. For example: skip 00:02:30 skip -170

If you execute a time jump while playback is paused or stopped, the TLogger retains that state. The TLogger also retains a playing state after a time jump unless the end of the file is reached and wrapping is disabled. For more information, see [section 3.6](#), “*End-of-File Action*”, and “*Controlling End-of-File Behavior in the TLogger*”, next.

When you use the **time** or **skip** command during slow-speed playback, the TLogger can perform *entity maintenance* to facilitate the transmission of up-to-date PDUs. For a description of entity maintenance, see “*Updating Entity Information Quickly After a Time Jump*” under [section 3.5.1](#). For information about how to enable or disable entity maintenance, see [section 5.5](#).

5.4.1. Controlling End-of-File Behavior in the TLogger

The TLogger normally enters a stopped state when it reaches the end of a tape. To specify a different end-of-tape action, start the Logger with the **-e** option:

```
TLogger -p filename -e action
```

Replace **action** with **p** to have the Logger enter a paused state at the end of a file, or **w** to wrap around to the other end of the file and continue playback.

5.5. Controlling Entity Maintenance in the TLogger

When you use a `time` or `skip` command during slow-speed playback, the TLogger's *entity maintenance* feature can ensure that up-to-date entity state PDUs are transmitted as soon as possible after the time jump. Entity maintenance is described in detail in [“Updating Entity Information Quickly After a Time Jump”](#) under section 3.5.1.

By default, the TLogger performs entity maintenance when you issue a `skip` command, but not when you issue a `time` command. To change these settings, issue the following commands at the TLogger prompt. Set *n* to 1 to enable entity maintenance, or 0 to disable it:

```
timemaint n
skipmaint n
```

To view the current `timemaint` or `skipmaint` setting, enter the command name by itself at the TLogger prompt.

5.6. Setting Playback Timescale and Direction

You can play a recording at any practical speed, forward or backward. To set the playback speed and direction, enter the `scale` command at the TLogger prompt:

```
scale timescale
```

where `timescale` is the factor by which time is to be multiplied. For information about choosing timescale values, see [section 3.4, “Playback Timescale and Direction”](#).

If desired, you can enter the timescale value in scientific notation, as in the following example for 10,000:

```
scale 1e+4
```

Note: Although the TLogger lets you specify a very large timescale (beyond 1e+35), the use of an unusually large timescale may produce unexpected results because of hardware limitations, and is not recommended.

If a recording is being played at a high speed, the TLogger may be unable to keep up, especially during passages that contain an unusually high concentration of data. If the TLogger falls significantly behind, it continues to process the PDUs as quickly as possible. No PDUs are discarded.

5.6.1. Setting an Initial Timescale at TLogger Startup

To specify an initial timescale, use the `-S` command-line option, for example:

```
TLogger -p my-exercise.lgr -s 2.5
```

5.6.2. PDU Processing for Non-Standard Timescales

When playing at speeds above or below normal, the TLogger makes corrective adjustments to the velocity and acceleration values in outgoing entity state PDUs. For details, see [section 3.5.2, “Velocity and Acceleration Scaling”](#).

The TLogger can generate interim PDUs during slow-speed playback to prevent receiving applications from timing entities out too soon. For more information, see [section 3.5.1, “Generating Interim PDUs During Slow-Speed Playback”](#), and [section 5.7, “Controlling Heartbeat and Timeout in the TLogger”](#).

5.7. Controlling Heartbeat and Timeout in the TLogger

During slow-speed playback, the TLogger generates interim PDUs as needed to maintain an appropriate heartbeat rate and timeout threshold for each entity in the recording. You can find an explanation of the heartbeat and timeout concepts in [“Heartbeat and Timeout”](#) under section 3.5.1.

The heartbeat rate determines how frequently the TLogger generates interim PDUs for each entity. By default, the TLogger maintains a heartbeat rate of five seconds. To increase or decrease this rate, use the `heartbeat` command at the TLogger prompt:

```
heartbeat n
```

where *n* represents the maximum allowable time between PDUs. To view the current `heartbeat` setting, enter the command name by itself.

The timeout threshold determines how long the TLogger continues to generate interim PDUs for an entity once the entity’s PDUs stop occurring in the recording. The TLogger’s default timeout setting is twelve seconds. To change this value, use the `timeout` command at the TLogger prompt:

```
timeout n
```

where *n* represents the number of recorded seconds the TLogger should wait before timing an entity out. To view the current `timeout` setting, enter the command name by itself.

5.8. Using a Script to Control Recording and Playback

A script is a plain text file which can control the TLogger in either playback or record mode. Scripts are especially useful for creating a prepared demo, or for automatically bypassing unwanted segments during tape playback.

A script can contain any TLogger run-time commands listed in Table 5-2. Create the script in any text editor, then redirect it to the Logger as in these examples:

```
TLogger -p my_exercise.lgr < my_play_script.txt
TLogger -r new_exercise.lgr < my_record_script.txt
```

At startup, the TLogger enters a playing or recording state by default, then immediately begins processing commands in the script. The Logger does not respond to keyboard input when running a script, so you cannot manually enter **quit** to stop execution. Use **Ctrl-C** instead.

Controlling Processing with the Wait Command

The **wait** command controls how long the TLogger runs in its current state before processing the next command in the script. The sample script below provides an example.

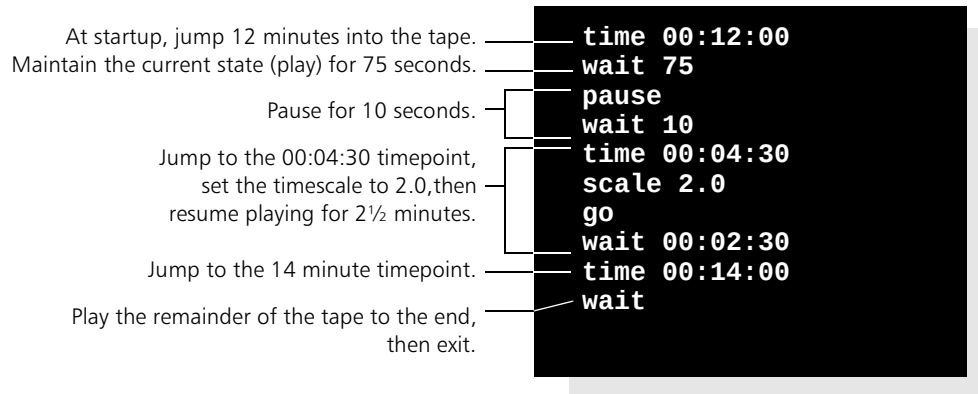


Figure 5-5. Sample TLogger script.

If **wait**, with a time argument, is the last command in a script, the TLogger maintains the current state for the duration of time argument, then exits.

If a script contains a **wait** command with no time argument, the TLogger continues operating in the current state until reaching the end of the tape, then exits. Any subsequent commands in the script are ignored.

5.9. Summary of TLogger Startup Options

Table 5-1: Command-line options for TLogger.

	Option	Description
Common	-h, -help, -?	Displays a brief summary of TLogger command-line options.
	-m <i>char</i>	Sets the TLogger's prompt character to <i>char</i> . Default is >.
	-P <i>dis_port</i>	Specifies which DIS port (UDP port) the TLogger uses to send or receive data. The default port is 3000.
Record Mode Only	-b <i>state</i>	Specifies whether the TLogger opens in a stopped state or begins recording immediately. Set <i>state</i> to r (TLogger default) for record, or s for stopped.
	-r <i>filename</i>	Causes the TLogger to open in record mode; specifies the destination file for recorded data. When recording with the TLogger, the -r switch and filename are required.
	-S <i>mcastAddr</i>	Subscribes the TLogger to the multicast address specified by <i>mcastAddr</i> . To subscribe to several multicast addresses, use this option more than once on the command line. For information, see section 3.7.2, "Subscribing to Multicast Groups" .
Playback Mode Only	-A <i>addr</i>	Sets the default IP address to which the TLogger sends PDUs. If this option is not used, the TLogger uses the computer's broadcast address. See section 3.7.1, "Specifying a Destination Address for Outgoing Data" .
	-d	Disables the generation of remote vehicle approximation (RVA) data when playing at speeds above or below normal. See section 3.5.2, "Velocity and Acceleration Scaling" .
	-e <i>action</i>	Specifies the TLogger's action upon reaching the end of the file. Set <i>action</i> to p to pause at the file's end, w to wrap to the other end of the tape and continue, or s to stop (default).
	-o <i>state</i>	Specifies whether the TLogger opens in a stopped state or begins playback immediately. Set <i>state</i> to p for playing, or s for stopped (TLogger default).
	-p <i>filename</i>	Causes the TLogger to open in playback mode and specifies the source file to be played. When playing back with the TLogger, the -p switch and filename are required.
	-S <i>timescale</i>	Sets the initial playback timescale factor. For more information, see section 3.4, "Playback Timescale and Direction" .
	-T <i>timepoint</i>	Sets the timepoint in the file where playback begins. Enter the <i>timepoint</i> in HH:MM:SS format. Default is 00:00:00

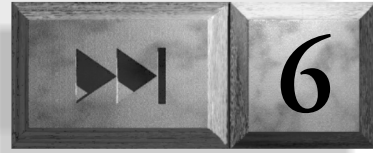
5.10. Summary of TLogger Run-Time Commands

Table 5-2: TLogger Run-Time Commands. (Part 1 of 2)

	Play	Record	Command	Description
Basic Controls	✓	✓	go	Starts or resumes recording or playback from a previously stopped or paused state.
	✓		pause	Pauses playback at the current timepoint and begins repeated resending of each entity's most recent entity state PDU, with velocity and acceleration set to zero.
	✓	✓	stop	Halts playback at the current timepoint, stops PDU recording or transmission.
	✓		scale [<i>factor</i>]	Sets the timescale factor that determines playback speed and direction. Use 1.0 (default) for normal speed, 2.0 for double speed, 0.5 for half speed, and negative values for backward playback.
	✓	✓	quit	Closes the TLogger file and exits the Logger.
Navigation	✓		start	Sets the current timepoint to the beginning of the recording.
	✓		end	Sets the current timepoint to the end of the recording.
	✓		time <i>time-point</i>	Navigates to a specific timepoint in the recording. Specify timepoint in HH:MM:SS format.
	✓		skip <i>offset</i>	Moves the current timepoint forward or backward by the specified time <i>offset</i> . Specify the offset in HH:MM:SS format or as a number of seconds. A positive offset causes a jump forward in time; a negative offset causes a backward jump.
PDU Generation and Processing	✓		timemaint [<i>n</i>]	Determines whether the TLogger generates up-to-date entity state PDUs after you jump to a new timepoint via the time command. Use a setting of 0 (default) to disable, 1 to enable, or no argument to display the current setting.
	✓		skipmaint [<i>n</i>]	Determines whether the TLogger generates up-to-date entity state PDUs after you jump to a new timepoint via the skip command. Use a setting of 0 to disable, 1 (default) to enable, or no argument to display the current setting.
	✓		heartbeat [<i>sec</i>]	During slow-speed playback, specifies the number of seconds the TLogger allows before generating an interim PDU for a given entity. Default is 5. With no argument, heartbeat displays the current setting.
	✓		timeout [<i>sec</i>]	During slow-speed playback, specifies how long the TLogger continues to transmit interim PDUs for an entity after the entity's PDUs have stopped occurring in the recording. Defaults to 12 recorded seconds.

Table 5-2: TLogger Run-Time Commands. (Part 2 of 2)

	Play	Record	Command	Description
Other	✓	✓	info or i	Displays a brief status report including the log file name, current timepoint, timescale, and current play state.
	✓	✓	h, help, or ?	Displays a summary of TLogger run-time commands.
	✓	✓	wait <i>time</i>	Used only in scripts. Specifies how long the TLogger performs its current action before processing the next command in the script. Enter <i>time</i> in the HH:MM:SS format or as a number of seconds.



Logger Tools

Chapter Overview

The Logger Tools are utilities that let you combine, modify and inspect recorded Logger tapes. This chapter describes:

- ♦ How to display the contents of a Logger file in a readable form with the **lgrDump** utility (page [6-2](#)).
- ♦ Using **lgrCat** to append or concatenate a series of Logger files together into a single file ([6-4](#)).
- ♦ Using **lgrMerge** to combine one or more logger files together into a single file, with all PDUs sorted in chronological order ([6-6](#)).
- ♦ Using **lgrOffset** to add or subtract a specified number of seconds to the time of each PDU in a Logger file ([6-7](#)).
- ♦ How to trim packets from the beginning or end of a file using **lgrTrim** ([6-8](#)).

Like the Logger itself, the Logger Tools are provided in separate versions for DIS revisions 2.0.3 and 2.0.4/IEEE-1278.1. For more information, please see [section 3.1.3](#), “Supported DIS Revisions”.

6.1. Viewing Logger Files with LgrDump

The **lgrDump** utility presents the contents of a Logger file in a readable form. Figure 6-1 shows a sample PDU as displayed by **lgrDump**. The output format is very similar to that of the **netdump** utility included with the VR-Link Toolkit.

```
*****
***** PDU #1077   Size=144           39.016 secs *****
*****
PduKind:           EntityStatePduKind   (1)
Version:           4
Exercise:          1
ProtocolFamily:    FamilyEntityInteraction   (1)
TimeStamp:         3297.213(seconds past hour)
TimeStampType:     Relative
Size:              144(bytes)
From:              1:15:1   (site:application:entity)
ForceID:           1
NumArtParams:      0
NumAttachedParts:  0
Type:              1:2:225:2:1:0:0 (kind:domain:country:cat:subcat:specific:extra)
Guise:             1:2:225:2:1:0:0 (kind:domain:country:cat:subcat:specific:extra)
Location:          { -2661471.288, -4358184.189, 3809002.286}
Velocity:          { 37.102, 87.402, 127.072}
Orientation:       { 1.169, -0.929, -2.305}
Acceleration:      { 0.000, 0.000, 0.000}
AngularVel:        { 0.000, 0.000, 0.000}
DrAlgorithm:       DrDrmRvw   (4)
Appearance:        0x00000000
  PaintScheme       _PaintSchemeUniform
  MobilityKill       _MobilityKillNone
  FirePowerKill      _FirePowerKillNone
  Damage             _DamageNone
  Smoke              _SmokeNone
  TrailingEffects    _TrailingEffectsNone
  Hatch              _HatchNA
  Lights              _LightsNone
  Flaming             _FlamingNo
  FinalPdu           _NotFinalPdu
  AfterBurner        UnknownEnum
Marking:            A10
CharSet:            1
Capabilities:       0x00000000
```

Figure 6-1. Sample PDU displayed by LgrDump.

Each PDU description begins with a heading containing a PDU number, packet size, and time elapsed since the start of the Logger tape. This information is added by the Logger during recording, and is not a part of the actual PDU.

After the heading, **lgrDump** displays all fields of the PDU.

6.1.1. How LgrDump Displays Unexpected PDU Types

This package provides separate **lgrDump** versions for DIS 2.0.3 and 2.0.4/IEEE-1278.1-1995. Each **lgrDump** executable displays formatted data for PDUs of its corresponding DIS version. If **lgrDump** encounters a PDU of another DIS version (or a non-DIS PDU), it displays available header information followed by a hex dump of the rest of the PDU.

6.1.2. LgrDump Usage and Examples

Usage:

```
lgrDump [filename]
```

Because **lgrDump** directs its output to standard output, you can redirect the output to a destination file or program as in these examples:

```
lgrDump logfile.lgr > output.txt  
lgrDump logfile.lgr | more
```

To pipe output from another program to **lgrDump**, omit the *filename* argument as in these examples:

```
lgrMerge File1.lgr File2.lgr | lgrDump  
cat exercise_1.lgr | lgrDump
```

Note: Owners of the VR-Link Toolkit can filter PDUs by specified field values by piping **lgrDump**'s output through the Toolkit's **buffgrep** utility. See the *VR-Link User's Guide* for more information.

6.2. Concatenating Files with LgrCat

The **lgrCat** utility concatenates together the PDUs from a series of two or more Logger files. The resulting file contains all PDUs from the first file followed by all PDUs from the second file, and so on.

In the output file, PDUs from the first file retain their original times and PDU numbers. For each subsequent file, **lgrCat** increases the starting time and PDU numbers to create a continuous PDU sequence, as in Figure 6-2.

If File1.lgr contains: PDU #1 at time 5.0 PDU #2 at time 10.0 PDU #3 at time 20.0	...and File2.lgr contains: PDU #1 at time 2.0 PDU #2 at time 4.0 PDU #3 at time 6.0
--	--

...then the concatenated file will contain:

```

PDU #1 at time 5.0
PDU #2 at time 10.0
PDU #3 at time 20.0
PDU #4 at time 22.0
PDU #5 at time 24.0
PDU #6 at time 26.0
    
```

Figure 6-2. LgrCat Example.

LgrCat modifies only the PDU numbers and time values generated by the Logger (those that appear in the PDU heading when viewed with **lgrDump**). **LgrCat** does not change the TimeStamp values contained in the original PDUs.

6.2.1. LgrCat Usage and Examples

Usage:

```
lgrCat [-s spacing] file1 [file2 file3...]
```

LgrCat directs its output to standard output (usually the display) in a fashion similar to the UNIX **cat** or DOS **type** command. In most cases, you redirect or pipe the output to a destination file or program, and specify two or more files to be concatenated, as in these examples:

```
lgrCat file1 file2 [file3 file4...] > output.file
lgrCat exercise1.lgr exercise2.lgr > ex_1-2.lgr
lgrCat ex1.lgr ex2.lgr ex3.lgr | lgrDump | more
```

6.2.2. Inserting Time Between Concatenated Files

You can use the optional `-s spacing` argument to insert a period of time between files. The *spacing* argument specifies the number of seconds to insert between the final PDU in each file and the starting point of the next file (timepoint 00:00:00). For example, the command,

```
lgrCat -s 4.0 File1.lgr File2.lgr > combined.lgr
```

creates a `combined.lgr` file whose first three PDUs match those in the output file in Figure 6-2. Now, however, the last three PDUs have been delayed by four seconds and now occur at 26.0 28.0 and 30.0 seconds, as in Figure 6-3:

If File1.lgr contains:	...and File2.lgr contains:
PDU #1 at time 5.0	PDU #1 at time 2.0
PDU #2 at time 10.0	PDU #2 at time 4.0
PDU #3 at time 20.0	PDU #3 at time 6.0

...then the concatenated file will contain:

PDU #1 at time 5.0
PDU #2 at time 10.0
PDU #3 at time 20.0
(four-second space)
PDU #4 at time 26.0
PDU #5 at time 28.0
PDU #6 at time 30.0

Figure 6-3. Effect of the LgrCat Spacer Argument.

6.2.3. Processing Input from Other Programs

To pipe the output of another program to **lgrCat**, replace one of **lgrCat**'s input file arguments with a hyphen character to specify where the incoming data should appear in the concatenated output.

In the example below, **lgrCat** receives output piped from the **lgrOffset** program as its first input (indicated by the hyphen), and concatenates that data with `File2.lgr`. The combined data is directed to `Out.lgr`:

```
lgrOffset 20.0 < File1.lgr | lgrCat - File2.lgr > Out.lgr
```

6.3. Merging Files with LgrMerge

The **lgrMerge** utility combines one or more logger files into a single file containing all PDUs from all files, sorted in chronological order and renumbered. Figure 6-4 provides an example:

If File1.lgr contains:	...and File2.lgr contains:
PDU #1 at time 1.0	PDU #1 at time 2.0
PDU #2 at time 3.0	PDU #2 at time 4.0
PDU #3 at time 5.0	PDU #3 at time 6.0

...then the merged file will contain:

```
PDU #1 at time 1.0
PDU #2 at time 2.0
PDU #3 at time 3.0
PDU #4 at time 4.0
PDU #5 at time 5.0
PDU #6 at time 6.0
```

Figure 6-4. How LgrMerge Combines Files.

LgrMerge modifies only the PDU numbers and time values generated by the Logger (those that appear in the PDU heading when viewed with **lgrDump**). **LgrMerge** does not change the TimeStamp values contained in the original PDUs.

6.3.1. LgrMerge Usage and Examples

Usage:

```
lgrMerge file1 file2 [file3 file4...]
```

Because **lgrMerge** directs its output to standard output, you typically redirect or pipe its output to a destination file or program as in these examples:

```
lgrMerge file1 file2 file3 > outputfile
```

```
lgrMerge exercise1.lgr exercise2.lgr > ex_1-2.lgr
```

```
lgrMerge ex1.lgr ex2.lgr ex3.lgr | lgrDump > PDUs.txt
```

6.4. Using *LgrOffset* to Shift PDUs in Time

The **lgrOffset** utility can adjust the time of all PDUs in a file by a specified number of seconds. You can:

- ♦ Decrease or increase the timepoints of all PDUs in a file so they occur earlier or later.
- ♦ Remove an unknown length of dead time from the beginning of a file.

During recording, the Logger records the arrival time of each PDU relative to the start of recording. This value is the only one that **lgrOffset** modifies; it is not a part of the original PDU. The original TimeStamp values in the PDUs are not affected.

6.4.1. *LgrOffset* Usage and Examples

Usage:

```
lgrOffset time|first
```

where *time* is the desired time offset in seconds. *time* can be a positive or negative value.

To remove a period of dead time from the beginning of a file, use the **first** keyword as the argument instead of a numeric *time* value. This causes **lgrOffset** to determine the timepoint of the first PDU, and to subtract that number of seconds from all PDUs in the file.

Because **lgrOffset** receives its input from standard input and directs its output to standard output, piping or redirection are used to specify the input source and output destination.

In either of the following two examples, **lgrOffset** increases the timepoints of all PDUs in *infile.lgr* by 15 seconds and saves the output in *outfile.lgr*:

```
lgrOffset 15.0 < infile.lgr > outfile.lgr
lgrCat infile.lgr | lgrOffset 15.0 > outfile.lgr
```

In the next example, **lgrOffset** removes an unknown amount of dead time from the beginning of *infile.lgr* and saves the output in *outfile.lgr*:

```
lgrOffset first < infile.lgr > outfile.lgr
```

6.5. Using LgrTrim to Extract Part of a File

The **lgrTrim** utility reads and outputs a specified portion of a Logger file, in effect, trimming packets from the beginning and end of the file. You can specify the part of the file to be output, as either a range of PDUs or a span of time.

6.5.1. LgrTrim Usage and Examples

Usage:

```
lgrTrim [-n] start [end]
```

The *start* and *end* arguments determine the part of the file to be included in the output. **LgrTrim** discards any packets that fall outside of the specified range. If no *end* is specified, **lgrTrim** outputs all packets from the specified *start* point to the end of the file.

By default, **lgrTrim** assumes that the *start* and *end* arguments are time values, given in seconds. To specify a numerical range of PDUs as the *start* and *end* points, use the *-n* option.

Because **lgrTrim** receives its input from standard input and directs its output to standard output, piping or redirection are used to specify the input source and output destination.

In the example below, **lgrTrim** extracts from *infile.lgr* all PDUs that occur between times 00:00:05 and 00:00:10 and saves them in *outfile.lgr*:

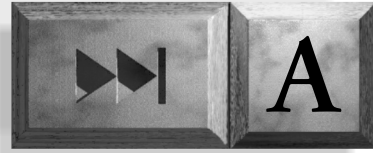
```
lgrTrim 5.0 10.0 < infile.lgr > outfile.lgr
```

Below, **lgrTrim** extracts the 10th through 20th PDUs from *infile.lgr* and saves them in *outfile.lgr*:

```
lgrTrim -n 10 20 < infile.lgr > outfile.lgr
```

The final example shows **lgrTrim**'s output piped to **lgrDump** for on-screen display of the 30th through 35th PDUs in *exerc01.lgr*:

```
lgrTrim -n 30 35 < exerc01.lgr | lgrDump
```



Troubleshooting

Chapter Overview

This chapter describes how to identify and correct the most common network problems that prevent the Logger from sending and receiving data properly:

- ♦ Ensuring the Logger uses the correct IP address and UDP port ([A-2](#))
- ♦ Specifying the correct network mask in Windows ([A-3](#))
- ♦ Correcting or bypassing hostname problems in Windows 3.1x ([A-3](#))

1.1. Troubleshooting Logger Network Communications

In a typical DIS exercise, VR-Link applications communicate with other applications using broadcast UDP/IP over a local area network. All applications in the exercise use the same IP broadcast address, and each application's host machine has a network interface device that is configured for that address. All machines must share a common netmask.

In addition, all participating applications must use the same UDP port.

If the Logger uses an IP address, netmask or port different from that used by your other applications, communication problems occur.

1.1.1. Using the Correct Broadcast Address and UDP Port

By default, the Logger communicates to the host machine's default broadcast address, using UDP port 3000. If the Logger appears to start normally, but fails to communicate with other DIS applications, the Logger may be using an incorrect broadcast address or UDP port. You can correct this problem by running the Logger with arguments that explicitly specify the correct address and port:

1. Determine the broadcast address used by your other DIS applications.

This may be the network's default broadcast address, or may be specified through the application's command line or configuration file at startup. For more information, see your System Administrator, or use the UNIX *ifconfig* command to display the broadcast address.

2. Determine the UDP port used by your other applications.
3. Start the Logger and notice the status message that appears at startup:

```
NetSocket ok for port 3000 to 123.456.789.255
```

4. If the port and address in the status message do not match those used by your other applications, exit the Logger and restart using the *Address* option (-A), the *DIS port* option (-P), or both, to specify the appropriate values. Enter:

```
Logger -A address -P portnum
```

where *address* and *portnum* are the broadcast address and UDP port used by your other applications, for example,

```
Logger -A 123.456.999.255 -P 3020
```


1.1.2. Configuring the Network Mask (Windows only)

The Windows versions of the Logger and TLogger expect to run on a class C network using a netmask of 255.255.255.0. If your machine is configured to use a different netmask, create a *DtNetmask* environment variable to reflect your netmask.

The DtNetmask variable should specify your netmask in a C-language style hexadecimal format. For example:

If your netmask is:	255.255.0.0
the hexadecimal representation is:	0xFFFF0000

The default value of DtNetmask is 0xFFFFF00, which corresponds to the netmask 255.255.255.0.

To set the DtNetmask variable for Windows NT:

1. Start the Windows NT Control Panel. Open the *System* applet.
2. Enter the DtNetmask variable name and appropriate hexadecimal value in the fields at the bottom of the *System* dialog box, then click *Set*.

To set the DtNetmask variable for Windows 95 or Windows 3.1x:

1. Add the following line to your AUTOEXEC.BAT file, then reboot. Replace the hexadecimal number with one that represents your netmask:

```
SET DtNetmask=0xFFFF0000
```

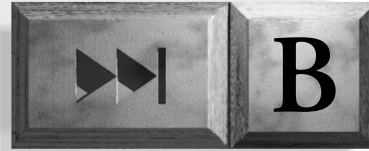
1.1.3. Hostname Table Errors (Windows 3.1x only)

If the Logger aborts with the message:

```
Cannot retrieve IP address of host.  
This could be a hosttable problem.
```

the PC may have an improperly configured hostname table. To test for this condition, have the PC ping itself by name. If the ping program returns a not-found message or reports a bad IP address, the PC's name and address are probably incorrect or missing from the table.

To solve this problem, edit your computer's entry in the hostname table (see your System Administrator or Windows documentation for information). Alternately, you can bypass the hostname table by running the Logger with the *-A* command line option, thereby setting the broadcast address explicitly.



Logger Manual Pages

This appendix contains printouts of the following Logger on-screen manual pages:

- ♦ LgrTools (page B-2)
- ♦ TLogger (B-4)
- ♦ Logger (B-7)

The LgrTools page covers the following utility programs:

- ♦ lgrCat (page B-2)
- ♦ lgrDump (B-3)
- ♦ lgrMerge (B-3)
- ♦ lgrOffset (B-3)
- ♦ lgrTrim (B-3)

lgrTools

Name

lgrCat, **lgrDump**, **lgrMerge**, **lgrOffset**, **lgrTrim** - logger file editing tools

Synopsis

```
lgrCat [-s spacing] file1 [file2 file3 ...]
lgrDump [filename]
lgrMerge file1 file2 [file3 file4 ...]
lgrOffset time
lgrTrim [-n] start [end]
```

Description

lgrCat, **lgrDump**, **lgrMerge**, **lgrOffset** and **lgrTrim** are off-line tools used to edit, combine or inspect Logger files. These utilities write their output to stdout.

lgrCat concatenates the PDUs from several Logger files together. The resulting Logger file contains all PDUs from *file1* followed by all PDUs from *file2*, *file3* and so on.

lgrCat offsets each file's starting timepoint ($t = 0$) so that it follows the time of the final PDU in the previous file. For example, if *File1.lgr* has 3 PDUs, one each at 5.0, 10.0 and 20.0 seconds, and *File2.lgr* has 3 PDUs, one each at 2.0, 4.0 and 6.0 seconds, then the logger file *Combined.lgr* resulting from

```
lgrCat File1.lgr File2.lgr > Combined.lgr
```

will contain 6 PDUs, one each at 5.0, 10.0, 20.0, 22.0, 24.0 and 26.0 seconds.

The optional *-n spacing* argument can be used to inset a specified number of seconds between the time of the final PDU in each file and the starting point of the successive file. For example,

```
lgrCat -n 3.0 File1.lgr File2.lgr > FileSpaced.lgr
```

produces *FileSpaced.lgr* whose first 3 PDUs are identical to those in *Combined.lgr*, but whose last 3 PDUs now occur three seconds later, at times 25.0, 27.0 and 29.0.

To use *stdin* as an input source for **lgrCat**, replace one of the file arguments with a hyphen character, for example,

```
lgrOffset 20.0 < File1.lgr | lgrCat - File2.lgr > Out.lgr
```

lgrDump converts the contents of a Logger file to readable form. Its output is similar to that of **netdump**, and may be piped through **buffgrep**. (These two utilities are part of the VRLink Toolkit.)

lgrDump accepts the name of a Logger file as input, and directs its output to *stdout* as shown:

```
lgrDump InFile.lgr > OutFile.txt
```

To use *stdin* as an input source for **lgrDump**, omit the filename argument, for example,

```
lgrMerge File1.lgr File2.lgr | lgrDump
```

lgrMerge merges together several Logger files to produce a file containing all PDUs from all files, sorted in chronological order. For example, if File1.lgr contains 3 PDUs, one each at 1.0, 3.0 and 5.0 seconds, and File2.lgr contains 3 PDUs, one each at 2.0, 4.0, and 6.0 seconds, then the logger file MergedFile.lgr resulting from

```
lgrMerge File1.lgr File2.lgr > MergedFile.lgr
```

contains 6 PDUs, one occurring at each second from 1.0 to 6.0 seconds.

lgrOffset is a filter that offsets each PDU in the input by a specified number of seconds. The *time* argument may be either positive or negative. If you specify the keyword **first** as the *time* argument, **lgrOffset** uses the negated time of the first PDU in the input file as the offset. This eliminates dead time from the beginning of a Logger file, and moves the first PDU to *t* = 0. **lgrOffset** takes its input from *stdin*:

```
lgrOffset 15.0 < original.lgr > offset.lgr
```

```
lgrOffset first < deadtime.lgr > nodeadtime.lgr
```

lgrTrim is a filter that trims packets from the beginning and/or end of the input. **lgrTrim** receives its input from *stdin* and writes all PDUs between *start* and *end* to *stdout*. Any PDUs outside the specified range are discarded. If the *end* argument is omitted, no PDUs are trimmed from the end of the input.

The *-n* option causes **lgrTrim** to trim on the basis of PDU count rather than time. For example, the file Trimmed.lgr created by

```
lgrTrim -n 5 10 < Whole.lgr > Trimmed.lgr
```

contains just the 5th through 10th PDUs from Whole.lgr. Without the *-n* option, PDUs are trimmed on the basis of time, so that

```
lgrTrim 5.0 10.0 < Whole.lgr > Trimmed.lgr
```

produces a Trimmed.lgr that contains the PDUs from Whole.lgr that occur between *t* = 5.0 and *t* = 10.0 seconds.

See Also

Logger, TLogger, buffgrep, netdump

TLogger

Name

TLogger - a DIS data logger and player with text-based shell-style interface

Synopsis

```
TLogger -h | -? | -help  
TLogger [options]
```

Description

The TLogger records a DIS packet stream to a file, or plays back such a file by sending out the recorded DIS packet stream.

In record mode, the TLogger can be in a recording state or a stopped state. In the recording state, the TLogger writes PDUs to the output log file. In the stopped state, the TLogger does not write PDUs to the log and does not advance the log's time.

In playback mode, the TLogger can be in a stopped, paused, or playing state. In the stopped state, the TLogger does not transmit PDUs and does not advance time. In the paused state, the TLogger repeatedly retransmits each entity's most recent PDU, but does not advance time. In the play state, the TLogger transmits PDUs and advances time at the rate set by the time scale setting.

Options

-A *address*

Sets the default IP *address* to which the TLogger sends PDUs. Defaults to the computer's broadcast address. (playback mode only)

-b *state*

Sets the TLogger's startup state when opened in record mode. Set *state* to "s" to open in a stopped state, or "r" (default) to begin recording immediately.

-d

Disables generation of remote vehicle approximation (RVA) data when playing back at non-normal speeds.

-e *action*

Specifies the TLogger's action upon reaching the end of the tape during playback. Set *action* to "p" to pause at tape end, "w" to wrap to the beginning of the tape and continue, or "s" to stop (default).

-h, -help, -?

Displays a brief summary of command line options, then exits.

-m *char*

Sets the TLogger's prompt character to *char*. Defaults to ">".

-o *state*

Sets the TLogger's startup state when opened in play mode. Set *state* to "s" to open in a stopped state, or "p" (default) to begin playing immediately.

-p *filename*

Causes the TLogger to open in play mode and play the specified file.

-P *dis_port*

Specifies which DIS port the TLogger uses for sending or receiving data. Default is 3000.

-r *filename*

Causes the TLogger to open in record mode and store the recording in the specified file.

-s *timescale*

Sets the initial timescale factor. A *timescale* of 1.0 (default) results in normal-speed playback. (playback mode only)

-S *mcastAddr*

Subscribe to the multicast address specified by *mcastAddr*. Can be used more than once on a command line to subscribe to multiple addresses. (record mode only)

-T *hh:mm:ss*

Sets the initial timepoint where playback begins. Default is 00:00:00

Commands

The following commands can be issued from the TLogger prompt during program execution. The (PO) marking denotes commands available in playback mode only.

go

Starts or resumes playback or recording from a stopped or paused state.

stop

Stops recording or playback at the current timepoint.

pause

Pauses playback at the current timepoint and begins repeated retransmission of each entity's most recent PDU. (PO)

wait *time*

Used in scripts to specify how long the TLogger performs its current action before processing the next command in the script. Enter *time* as HH:MM:SS or as a number of seconds.

start

Rewinds the Logger tape to the beginning. (PO)

end

Advances the Logger tape to the end. (PO)

time *timepoint*

Rewinds or advances the Logger tape to the *timepoint* specified in HH:MM:SS format. (PO)

skip offset

Rewinds or advances the tape by the specified positive or negative time *offset*, specified as HH:MM:SS or as a number of seconds. (PO)

timemaint *n*

Determines whether the TLogger updates entity information when the time command is issued. Set *n* to 0 (default) to disable, 1 to enable, or no argument to display the current setting. (PO)

skipmaint *n*

Determines whether the TLogger updates entity information when the skip command is issued. Set *n* to 0 to disable, 1 (default) to enable, or no argument to display the current setting. (PO)

scale factor

Sets the time scale *factor* that determines playback speed and direction. Default is 1.0 for normal speed forward. (PO)

heartbeat seconds

During slow-speed playback, specifies the number of seconds the TLogger allows before generating an interim PDU for a given entity. Default is 5. With no argument, heartbeat displays the current setting. (PO)

timeout seconds

During slow-speed playback, specifies how long the TLogger continues to transmit interim PDUs for an entity after the entity's PDUs have stopped occurring in the recording. Defaults to 12 recorded seconds. (PO)

info, *i*

Displays a brief status summary including the current log file name, timepoint, time scale, and record or play state.

h, help, ?

Displays a summary of TLogger commands.

quit

Closes the log file and exits the Logger.

See Also

Logger, VR-Link Logger User's Guide

Logger

Name

Logger - a Motif-based or Windows-based DIS data logger and player

Synopsis

```
Logger -h | -? | -help
Logger [options]
```

Description

The **Logger** records a DIS packet stream to a file, or plays back such a file by sending out the recorded DIS packet stream. The UNIX version has a Motif-based VCR-style graphical interface. The Windows version provides an identical interface and feature set, and runs in a Microsoft Windows or Windows NT environment.

In playback mode, the **Logger** can be in a stopped, paused, or playing state. In the stopped state, the **Logger** does not transmit PDUs and does not advance time. In the paused state, the **Logger** repeatedly retransmits each entity's most recent PDU, but does not advance time. In the play state, the **Logger** transmits PDUs and advances time at the rate set by the time scale setting.

The **Logger**'s playback mode control set includes buttons for Stop, Pause and Play, along with Start and End buttons that cause a jump to the beginning or end of the recording. A slider control sets the time scale and direction of playback.

In record mode, the **Logger** can be in a recording state or a stopped state. In the recording state, the **Logger** writes PDUs to the output log file. In the stopped state, the **Logger** does not write PDUs to the log and does not advance the log's time. The **Logger**'s record mode control set includes only two buttons, Record and Stop.

In either mode, the **Logger**'s pull-down menus provide access to a range of additional controls and options, summarized below.

Options

-A *address*

Sets the default IP *address* to which the Logger sends PDUs. Defaults to the computer's broadcast address.

-b *state*

Sets the Logger's action when a file is opened in record mode. Set *state* to "s" (default) to open in a stopped state, or "r" to begin recording immediately.

-d

Disables generation of remote vehicle approximation (RVA) data when playing back at non-normal speeds.

-e *action*

Specifies the Logger's action upon reaching the end of the tape during playback. Set *action* to "p" to pause at tape end, "w" to wrap to the beginning of the tape and continue, or "s" to stop (default).

-h, -help, -?

Displays a brief summary of command line options, then exits.

-o *state*

Sets the Logger's action when a file is opened in play mode. Set *state* to "s" to open in a stopped state (default), or "p" to begin playing immediately.

-p [*filename*]

Causes the Logger to open in play mode and optionally load a file for playback.

-P *dis_port*

Specifies which DIS port the Logger uses for sending or receiving data. Defaults to 3000.

-r [*filename*]

Causes the Logger to open in record mode and optionally open a file to receive recorded data.

-s *timescale*

Sets the initial playback timescale factor. A *timescale* of 1.0 (default) results in normal-speed playback.

-S *mcastAddr*

Subscribe to the multicast address specified by *mcastAddr*. Can be used more than once on a command line to subscribe to multiple addresses.

-t *list*

Specifies the set of values that appear on the Logger time scale slider control. Enter *list* as a series of comma-separated values without spaces.

-T *hh:mm:ss*

Sets the initial timepoint where playback begins. Default is 00:00:00

File Menu Options

Play Back Old File

Displays a dialog box where you choose an existing file to play back.

Record New File

Displays a dialog box where you choose a file to open for recording. You can specify a name for a new file or select an existing file to overwrite.

Quit

Close the current file and exit the program.

Options Menu Options

Configuration

Displays a dialog box that allows control of the following options:

Play Action. Determines whether the Logger enters a stopped or playing state (default) when a file is opened for playback.

Record Action. Determines whether the Logger enters a stopped or recording state (default) when a file is opened for recording.

End of File Action. Determines what action the Logger takes when the end of a tape is reached during playback. Options are Wrap, Pause, and Stop (default).

Heartbeat Value. During slow-speed playback, specifies the number of seconds the Logger allows before generating an interim PDU for a given entity. Default is 5 seconds.

Timeout Value. During slow-speed playback, specifies how long the Logger continues to transmit interim PDUs for an entity after the entity's PDUs have stopped occurring in the recording. Defaults to 12 recorded seconds.

Set Time

Displays a dialog box where you specify a timepoint. When you click OK, the Logger jumps forward or backward in the Logger tape to the specified timepoint.

Set Time Scale

Displays a dialog box where you can change the playback time scale.

Skip Time

Displays a dialog box where you specify a time offset. When you click OK, the Logger jumps forward or backward in time by the specified offset.

Help Menu Options

About Logger

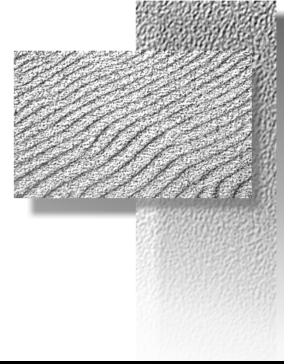
Displays product information pertaining to your release of the Logger.

Help

Displays a summary of Logger procedures and commands.

See Also

TLogger, VR-Link Logger User's Guide



Index

Symbols

-? option 4-10, 5-11
? command 5-13

A-B

-A option 3-10, 4-10, 5-11
acceleration scaling 3-9
appending Logger tapes 6-4
-b option 4-9, 4-10, 5-11
./bin and ./bin3 directories 2-4, 3-3
broadcast address A-2

C

case sensitivity of command line 3-2
command line options
 general format 3-2
 -? 4-10, 5-11
 -A 3-10, 4-10, 5-11
 -b 4-9, 4-10, 5-11
 -d 3-9, 4-10, 5-11
 -e 3-9, 4-9, 4-10, 5-6, 5-11
 for GUI Logger 4-10
 -h 4-10, 5-11
 -help 4-10, 5-11
 -m 5-11
 -o 4-9, 4-10, 5-11
 -P 4-10, 5-11
 -p 4-10, 5-11
 -r 4-10, 5-11
 -S 3-10, 4-10, 5-11
 -s 4-7, 4-10, 5-11
 and startup under Windows 4-3
 -T 4-10, 5-11
 -t 4-7, 4-10

concatenating Logger tapes 6-4
Configuration options in GUI 4-9

D

-d option 3-9, 4-10, 5-11
dead reckoning
 and acceleration scaling 3-9
 effect of reverse playback on 3-9
 and velocity scaling 3-9
dead time, removing from start of tape 6-7
destination address 3-10
Dial-a-Tank 1-3
directories
 ./bin 2-4
 ./bin3 2-4
 ./doc 2-4
 ./logger_tapes 2-4
DIS
 described 1-2
 supported versions 3-3
DIS debugging tools 1-3
DIS port
 Logger default A-2
 specifying 4-10, 5-11, A-2
disk space requirements 2-2
DtNetmask environment variable in
 Windows A-3

E-F-G

-e option 3-9, 5-6, 3-9, 4-9, 4-10, 5-11
editing a Logger tape 6-8
End button 4-6
end command 5-6, 5-12

- end-of-file action 3-9
 - in GUI 4-9
 - in TLogger 5-6
- entity maintenance 3-8, 4-6, 5-6, 5-7
- exiting Logger during script 5-10
- extracting part of a Logger tape 6-8
- go command 5-12

H-I-J

- h option 4-10, 5-11
- help option 4-10, 5-11
- h command 5-13
- heartbeat
 - defined 3-7
 - setting in GUI 4-9
 - TLogger command 5-12
- help command 5-13
- hostname error in Windows A-3
- i command 5-13
- info command 5-5, 5-13
- installation 2-3
- interim PDUs 3-7, 3-8, 4-8, 5-8

K-L

- lgrCat 6-4
 - effect on PDU timestamp data 6-4
 - pipng output from another program 6-5
- lgrDump 6-2
 - compared with netdump 6-2
 - display of unexpected PDU types 6-3
 - using with buffgrep utility 6-3
- lgrMerge 6-6
- lgrOffset 6-7
- lgrTrim 6-8
- Logger
 - component files 2-4
 - distribution media 2-2
 - interfaces, compared 3-2
 - man page B-4, B-7
 - overview 3-2
 - software components of 1-1
 - UNIX vs. Windows versions 3-2

- Logger tapes
 - combining, using lgrMerge 6-6
 - concatenating with lgrCat 6-4
 - editing 6-8
 - extracting a portion of 6-8
 - installing 2-2
 - modifying time data with lgrOffset 6-7
 - removing dead time from beginning 6-7
 - samples 3-3
 - viewing PDU contents of 6-2
 - viewing with Stealth 3-3
- Logger Tools
 - described 6-1
 - for specific DIS versions 6-1
 - man page B-2

M-N-O

- m option 5-11
- MäK product line 1-3
- MäK Stealth 1-3
- man pages B-1
 - lgrTools B-2
 - Logger B-7
 - TLogger B-4
 - viewing as text files in Windows 3-2
- multicast address 3-10
- network mask (netmask) A-3
- network, requirements for Logger A-2
- o option 4-9, 4-10, 5-11
- overview 3-2

P

- P option 4-10, 5-11
- p option 4-10, 5-11
- pause
 - compared to stop 5-4
- pause command 5-12
- PDU
 - defined 1-2
 - how modified in Logger tape 6-2
 - viewing with lgrDump 6-2
- platforms, supported 2-2
- play action
 - in GUI 4-9
 - in TLogger 5-5

- playback 3-5
 - described 3-5
 - in GUI 4-5
 - in reverse 3-6
 - in TLogger 5-4
 - playback direction
 - in GUI 4-7
 - in TLogger 5-8
 - playback mode 3-5
 - playback speed
 - allowable range of 4-8, 5-8
 - in GUI 4-7
 - in TLogger 5-8

Q-R

- quit command 5-12
- r option 4-10, 5-11
- record action
 - in GUI 4-9
 - in TLogger 5-3
- record mode 3-4, 5-2
- recording 3-4
 - described 3-4
 - with GUI 4-4
 - status, displaying in TLogger 5-3
 - with TLogger 5-2
- red timescale display 4-8
- reverse playback
 - and dead reckoning 3-9

S

- S option 3-10, 4-10, 5-11
- s option 4-7, 4-10, 5-11
- sample tapes 3-3
- scale command 5-8, 5-12
- scaling, time 3-9
 - in GUI 4-8
 - in TLogger 5-8
- script
 - creating 5-10
 - executing 5-10
 - and graphical interface 3-2
 - quitting Logger while running 5-10
 - and wait command 5-10
- Set Time 3-8
 - and entity maintenance 4-6
 - menu option 4-6
- Set Time Scale menu option 4-8

- skip command 5-6, 5-12
 - and entity maintenance 3-8, 5-6, 5-7
 - Skip Time 3-8
 - and entity maintenance 4-6
 - menu option 4-6
 - skipmaint command 3-8, 5-12
 - slow-speed playback 3-6, 5-8
 - Start button 4-6
 - start command 5-6, 5-12
 - startup
 - from DOS box 4-3
 - from icon under Windows 4-3
 - playback mode 3-5
 - record mode 3-4
 - TLogger 5-2
 - TLogger, record mode 5-2
 - UNIX 4-3
 - status field 4-4, 4-5
 - status message, in TLogger playback 5-5
 - Stealth 1-3
 - stop command
 - compared to pause 5-4
 - in playback mode 5-12
 - in record mode 5-3
 - supported platforms 2-2
 - system requirements 2-2

T-U

- T option 4-10, 5-11
- t option 4-7, 4-10
- tape formats 2-2
- time
 - between concatenated Logger tapes 6-5
 - navigating during playback in GUI 4-6
 - navigating during TLogger playback 5-6
 - shifting PDU times 6-7
- time command 5-6
 - and entity maintenance 3-8, 5-7
 - in TLogger 5-12
- time jump
 - updating entities after 3-8
- time navigation
 - in GUI 4-6
 - in TLogger 5-6
- time scale
 - described 3-6
 - setting in GUI 4-7, 4-8, 5-8
 - setting in TLogger 5-8
 - specifying at startup 4-7, 5-8

Index

- time scale slider control
 - controlling with keyboard 4-7
 - customizing 4-7
- timemaint command 3-8, 5-12
- timeout
 - defined 3-7
 - setting in GUI 4-9
 - setting in TLogger 5-12
- timepoint
 - setting in GUI 4-6
 - setting with TLogger skip command 5-6
 - setting with TLogger time command 5-6
- timescale 5-8
 - during recording 4-4
 - effects 3-7
 - red text, meaning of 4-8
- TLogger run-time commands
 - ? 5-13



MäK Technologies

*185 Alewife Brook Parkway
Cambridge, MA 02138
617.876.8085*